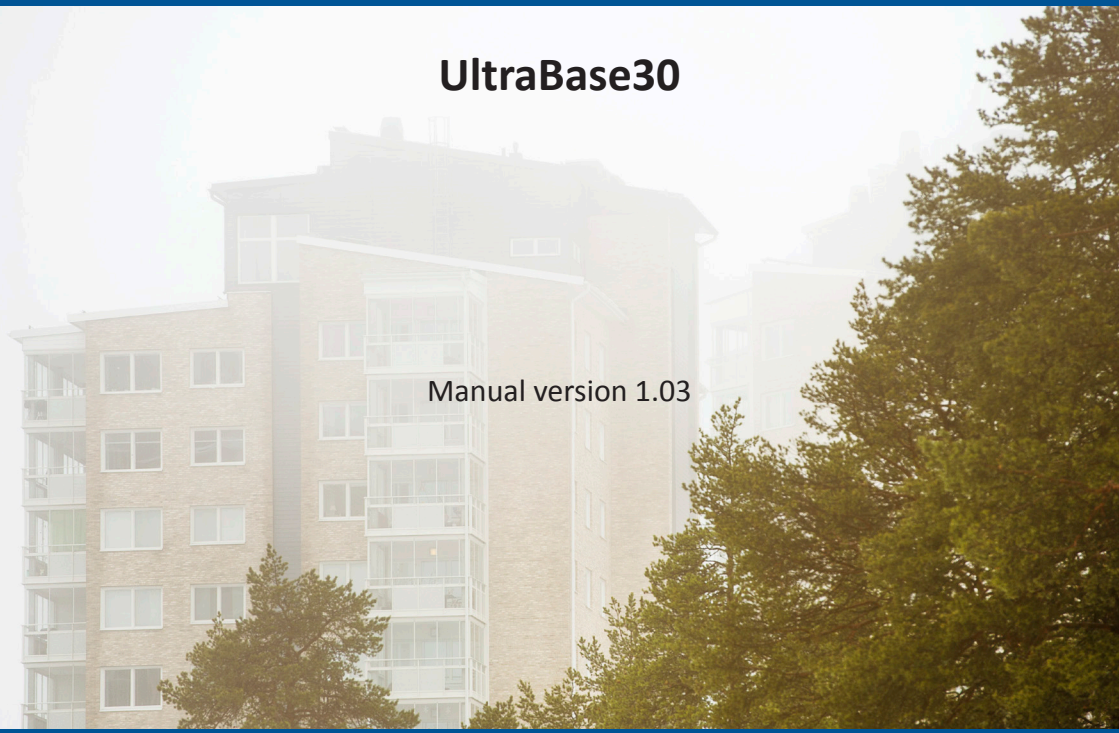
IMSE
Ultra

Configuration manual



UltraBase30

Manual version 1.03

Abelko INNOVATION

www.abelko.se

All information in this manual is based on information available at the time of printing. The manual is published to ease the use of an Ultra. Abelko Innovation cannot guarantee that there are no mistakes or errors in this documentation and cannot be held responsible for any consequences resulting from the use or misuse based on this information.

All information in this document can be changed without notice. It is likely that certain sections will be changed at the release of new product versions. Be sure to have the latest version of this document and the corresponding version of Ultra.

© Abelko Innovation. All Rights Reserved.

Contents

1 Introduction....1
1.1 Manual Version....2
1.2 Other Manuals....2
1.3 Important Information....2
1.3.1 Installation and Protection....2
1.3.2 Electromagnetic Radiation....2
1.4 Warranty....2
2 Start-up....4
2.1 Power Supply....5
2.2 Connect Sensors and Actuators....5
2.2.1 Universal Inputs....5
2.2.2 Digital Inputs....6
2.2.3 Analog Outputs....6
2.2.4 Digital Outputs....7
2.3 Connect Expansion Modules....7
2.3.1 Digital Outputs....7
2.4 Network Connection....8
2.5 Login and Password....9
2.6 Overview Web Interface....10
3 Applications & Resources....14
3.1 Applications....14
3.2 Channels....14
3.3 Alarms....14
3.4 Alarm Groups....15
3.5 Parameters....15
3.6 Curves....15
3.7 Time Schedule....15

4 Graphical Programming....16
4.1 Create Controllers....16
4.2 Create Graphical Programs....17
4.2.1 Create a Template of a Subapplication....24
4.2.2 Backup Applications....25
4.3 Channels....26
4.3.1 Channel logs....28
4.4 Alarms....29
4.5 Alarm Groups....31
4.6 Curves....32
4.7 Parameters....34
4.8 Time Schedules....35
4.9 Applications Status....42
4.10 Overview Icons Graphical Programming....43
4.11 Explanation of the Different Components....45
5 Overviews....50
5.1 Create a New Overview....50
5.1.1 Add Backgrounds....51
5.1.2 Add Resources....51
5.1.3 Add Text Boxes and Links....53
5.2 Tools....54
5.3 View Alternatives....54
6 Summaries....56
6.1 Create a New Summary....56
6.2 Summary Tools....59
6.3 Summary View Alternative....59
6.4 Channels....60
6.5 Alarms....60

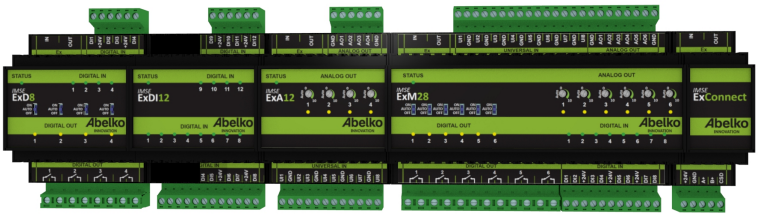
6.6 Time Schedules....61
6.7 Parameters....61
6.8 Curves....61
7 Alarms....62
7.1 Active Alarms....62
7.2 Alarm Types....62
7.3 Edit Alarms....63
7.4 Create Alarm Groups....65
7.5 Alarm Sendouts....66
7.6 Events/Errors and Messages....68
8 Data Logs....69
8.1 Smart Log....69
8.2 Periodic Log....69
8.3 Create logs....69
8.4 View Logs....70
9 Inputs & Outputs....72
9.1 Overview....72
9.2 Configure Sensors and Actuators....72
9.2.1 Configure Expansion Modules....74
9.3 Replacing an Expansion Module....76
10 External Units....78
10.1 Configure External Units....78
10.2 Status....79
11 Web shares....80
11.1 Create an Auto-export....80
11.2 Create an Export via I/O-unit....83
11.3 Create an Import via I/O-unit....85
11.4 Error Status....88

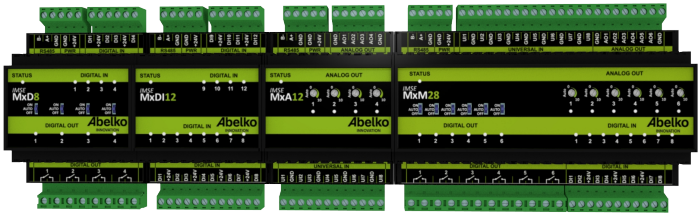
12 Manual Override....89
12.1 Allow Manual Override....89
12.2 Activate Manual Override....89
13 System....91
13.1 Notes....91
13.2 File Manager....91
13.3 Users....93
13.4 Settings....94
13.5 Backups and Updates....94
13.5.1 Via the Unit UltraBase30....95
13.6 Information....96
14 Communication....97
14.1 Network....97
14.2 Mail Host....97
14.3 Recipients....98
14.4 Modbus Slave Register....99
14.4.1 Register types....102
14.5 Modbus slave settings....103
14.6 Modbus TCP Gateway....104
14.7 Portal Update....105
14.7.1 Status....106
15 FAQ's....108
16 Technical Specifications UltraBase30....109
16.1 Universal Inputs....109
16.1.1 Accuracy....110
16.2 Digital Inputs....111
16.3 Analog outputs....111
16.4 Digital outputs....111

17 Term Definitions.....112

1 Introduction

Welcome to IMSE Ultra. This series of products consists of the control unit IMSE UltraBase30 and expansion modules that can be connected to expand the number of inputs and outputs.





This configuration manual is mainly for the person configuring the Ultra system. It aims to provide a guide on how to use the system in the easiest and most effective way.

1.1 Manual Version

Version 1.02, 2015-02-27: First draft of this manual.

Version 1.03, 2015-09-10: Second draft of this manual.

1.2 Other Manuals

User manual: a guide for how to use the configured system.

Reference manual: more detailed technical information and script programming.

Quick start guide: information on how to quickly get the system up and running.

All manuals are available for download at **www.ultra.abelko.se**.

1.3 Important Information

1.3.1 Installation and Protection

Electrical installation work may only be carried out by qualified electricians in accordance with the applicable regulations. **The equipment must be unplugged from the mains during installation**.

Ultra is protected to IP 21, which means it is not water resistant. It must **not be installed where it might be exposed to water of any kind**.

UltraBase30 is designed for an ambient temperature of between –20 °C and +50 °C and up to 90% relative humidity (non-condensing).

1.3.2 Electromagnetic Radiation

When measurement and control systems are installed, the cables must lie in a way that minimizes magnetic and electric fields. These fields can be influenced by many factors: frequency converters, relays, contactors, earth currents and static discharge. Cable lengths can also affect sensitivity. The installation should be carefully planned to minimize interference.

1.4 Warranty

1. Abelko will repair any design, material and manufacturing defects at its

own expense, provided they occur during normal use and the purchaser submits a claim within 60 months of the verified delivery date. The purchaser is responsible for removal, re-installation and for paying transport costs to Abelko, and Abelko will repair the defect and return the equipment free of charge to the purchaser.

2. The warranty only covers design, material and manufacturing defects. Abelko is not responsible for defects caused by a failure to follow the instructions, or defects resulting from normal wear and tear, poor maintenance, unauthorized work, non-compliant operating conditions, incorrect installation or repairs not carried out by Abelko or an authorized agent, voltage surges or other electrical faults.

3. Abelko's responsibility for defects is limited to the circumstances described above. Abelko is not responsible for any consequential damage that may occur as a result of design, material and manufacturing defects. The purchaser is therefore not entitled to use defects as grounds for compensation or any other claim, except in the circumstances described above, nor may such claim be made against any third parties responsible for fulfilling this warranty.

4. Abelko is not responsible for restoring any configurations, etc. added by the purchaser. The purchaser should create a backup of configurations and save them to a server.

The information in this document is the property of Abelko Innovation. The content is confidential and unless you have Abelko Innovation's written permission, you may not disclose it to anyone except Abelko Innovation personnel, resellers, agents or licensees. You may not copy parts of the document or save it to data media or other media, including photocopying or recording, unless you have the permission of the copyright holder, Abelko Innovation.

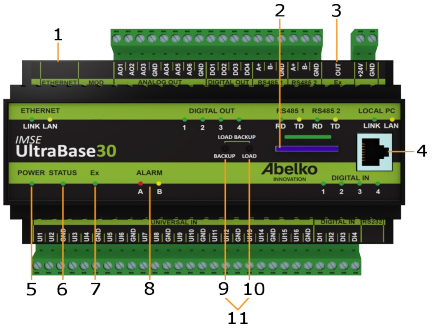
Abelko Innovation makes no guarantees in respect of the content of this document. Abelko Innovation also retains the right to alter, add or remove parts of the document at any time and without notice. Reasons for doing so include printing errors, incorrect information and software/product improvements. Any such changes will always be included in new editions of this document.

All rights reserved.

© Abelko Innovation 2015

2 Start-up

In this chapter, we will explain what you need to know to begin setting up the system.



1. **ETHERNET** is used to connect to the network.
2. Place for SD card.
3. **Ex OUT** is used to connect to expansions modules.
4. **LOCAL PC** is used only for direct connection to a PC with address **https:\\192.168.142.1**.
5. **POWER** shines green when the unit is powered.
6. **STATUS** shines green when the unit is running. It may take a little while before **STATUS** lights up when the unit is powered. If it does not light up, inputs and outputs are not working.
7. **Ex** shines green when all the expansion modules that should be connected are running.
8. **ALARM** has one red LED for **A-alarms** and one yellow LED for **B-alarms.**
9. **BACKUP** is used to create a full backup of the system to the SD card.
10. **LOAD** is used to load a file from the SD card into the unit. The file must be a zip file that can contain sequence files, clone backup (a backup without data bases), software updates, script of the parameter bank and more.
11. **LOAD + BACKUP** is used to load and run a full backup from the SD card to the unit.

2.1 Power Supply

UltraBase30 is powered by a 24 V **DC** power supply through the terminal at the top right (marked **+24V** and **GND**) and must supply at least 300 mA. The power supply also provides power to any expansion modules and must be dimensioned for them.

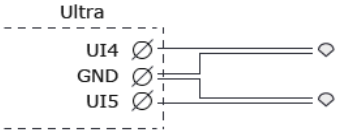
The **Power**-LED shines when the module has power. The **Status**-LED shines when the unit is running.

2.2 Connect Sensors and Actuators

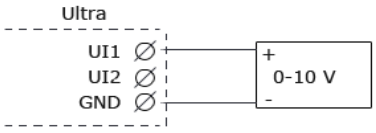
2.2.1 Universal Inputs

Universal inputs, marked **UI**, are located on the lower terminal strip. Universal inputs can be used for both resistive temperature sensors, voltage 0-10 V and current 0-10 V. UI can also be used as a digital status input for potential-free switches. You need to make settings for the various sensors. This is done under the menu **Configuration**, **In- & outputs**.

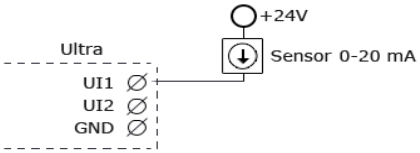
Resistive temperature sensors



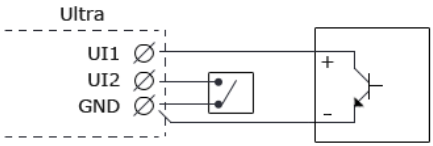
Voltage 0-10 V



Current 0-20 mA

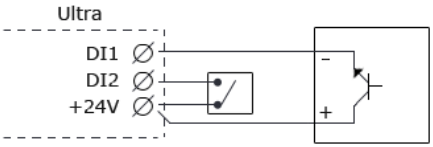


Digital status input for potential-free switches



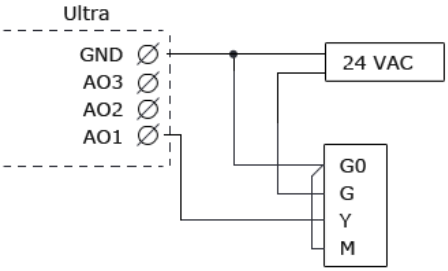
2.2.2 Digital Inputs

Digital inputs, marked **DI**, are located on the lower terminal strip. These can be used for digital status, counters and frequency.



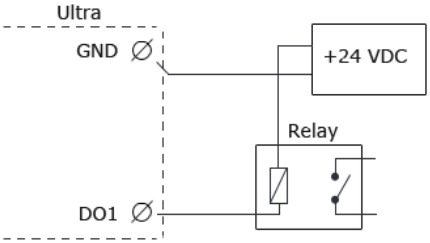
2.2.3 Analog Outputs

Analog outputs 0-10 V, marked **AO**, are located on the upper terminal strip. The following figure is an example on how you can connect an AC-fed valve controller.



2.2.4 Digital Outputs

Digital outputs, marked **DO**, are located on the upper terminal strip. Digital outputs are open collector that closes to earth (GND) when they are activated.



For more detailed information see chapter **Technical Specification**.

You can configure all of these sensors and actuators in the interface under the menu **Configuration** and **In- & outputs**. Read more about this in chapter **In- & Outputs**.

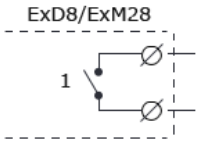
2.3 Connect Expansion Modules

Expansion modules are connected with a modular cable from **Ex OUT** on UltraBase30. These are configured through the web interface under the menu **Configuration** and **In- & outputs**; read more about this in chapter **In- & Outputs**.

Keep in mind that if the system is preconfigured before the installation, it is important that the expansion modules are placed in the same order as configured for.

2.3.1 Digital Outputs

The expansion modules have the same connections as UltraBase30, with the exception for **ExD8** and **ExM28**. They have relay outputs that is called **DO**.



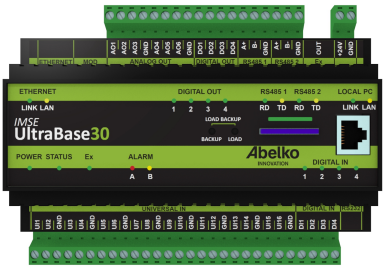
These relays are closing and have a breaking capacity of 250 VAC (2 A) and 250 VDC (2 A).

The **Ex**-LED shines green when all the expansion modules that should be connected is running.

2.4 Network Connection

To configure settings and access information in an Ultra, it must to be connected to a network or a computer.

UltraBase30



ETHERNET (top left)
This is used to connect to the fixed network.

LOCAL PC (right of the front)
This is used to connect the unit directly to a PC. The address is **https:\\192.168.142.1**.

The LOCAL PC port has a DHCP server. It provides a connected PC network setting, which gives direct access to web pages without manually changing the network settings. **This means that you should never connect this port to a fixed network.** The port is slower than the one labeled Ethernet, and should only be used to configure the regular network connection.

These have two LEDs each. **LINK** shines yellow when the Ultra is connected to a network, and **LAN** flashes green when there is communication on the network.

2.5 Login and Password

A new Ultra has the factory default IP address 10.0.48.94. Start a web browser and enter the address **https://10.0.48.94** in the address bar. This will take you to the login page.

Username and password the first time you log in:

Username **config**

Password **ef56**

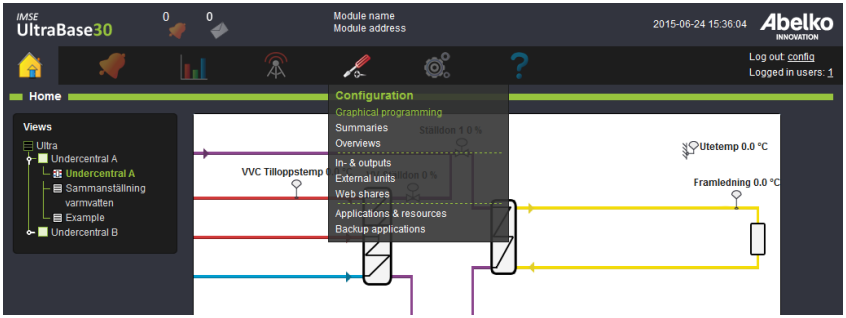
Warnings when logging in!

The first time you log in to an Ultra, you will be warned that the connection is not private, or similar. This is because the **HTTPS**, that is being used in the system, is a secure and encrypted connection. The security is guaranteed by a certificate, which needs to be issued for a specific IP number. Since the IP address is adjustable, there is no certificate for the unit. **You need to add this exception.** The procedure differs depending on which browser you are currently using.

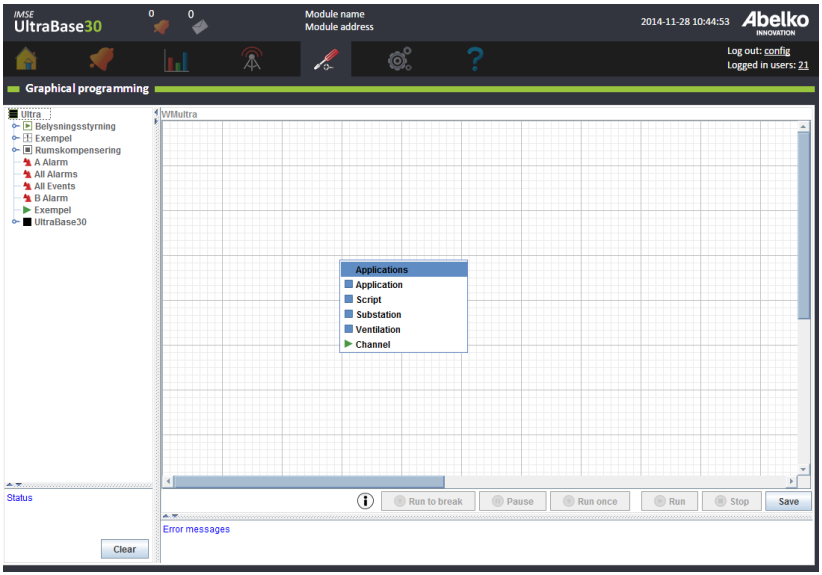
It is recommended that you change your password to maximize safety. To do this, go to **System** and **Users**. Here you can also change the password for other users and add new users.

You can change the network settings under the menu **Communication**, **Network**.

2.6 Overview Web Interface



The **graphical programming** can be found under the **Configuration** menu. Here you can configure substation and ventilation controllers, generate PID controllers and create applications with both graphical programming and script programming.



1. **Right-click** on the workspace and select the type of application you want to create. Substation, ventilation and PID take you to the controller tool.

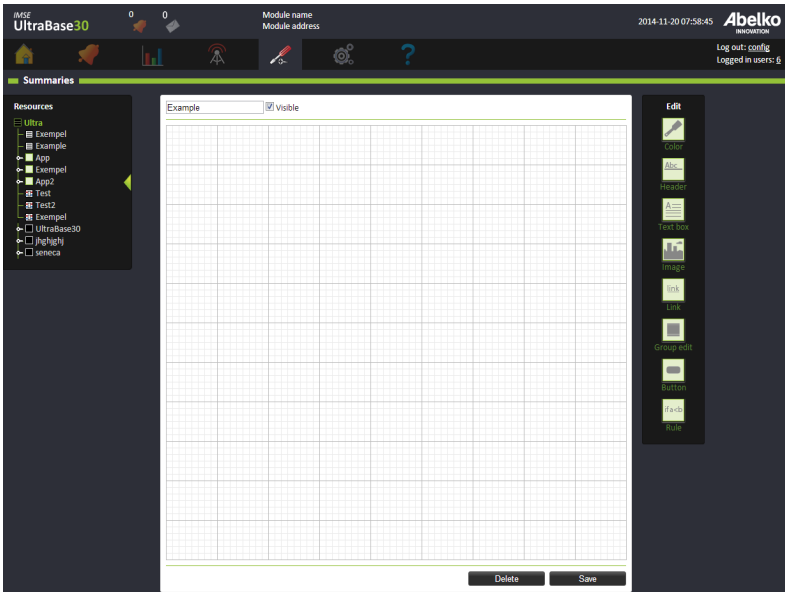
2. Add the controllers and applications you want to use.

3. Open an application by **double-clicking** on it. **Right-click** on the workspace inside an application to add components and resources.

4. Click **Save** down to the right.

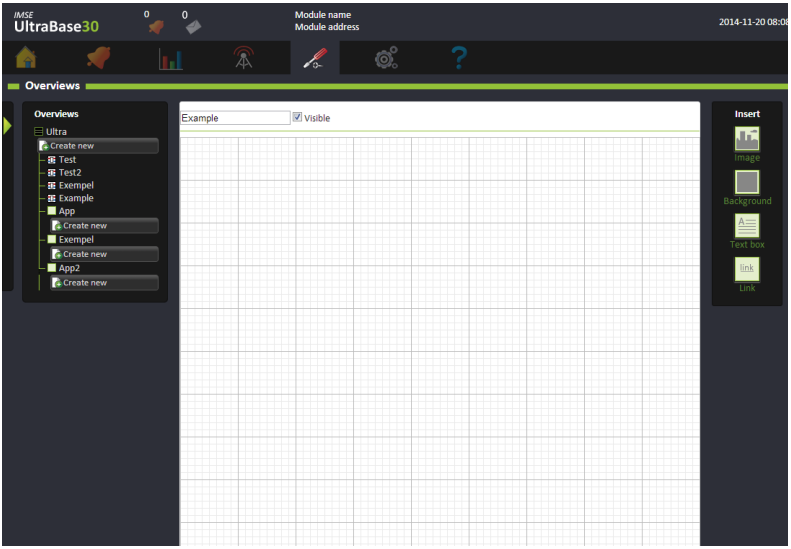
To understand how all the inputs, outputs etc. belong together, you may view the **node tree** (located to the left). The node tree contains all the elements in the system, arranged in hierarchical order. If you open an application, you can see what is in it, and further, if you fold out a subapplication, you can see what is in the subapplication.

Create **Summaries** and **Overviews** for a detailed overview of the system. These tools are also located under the menu **Configuration** menu.



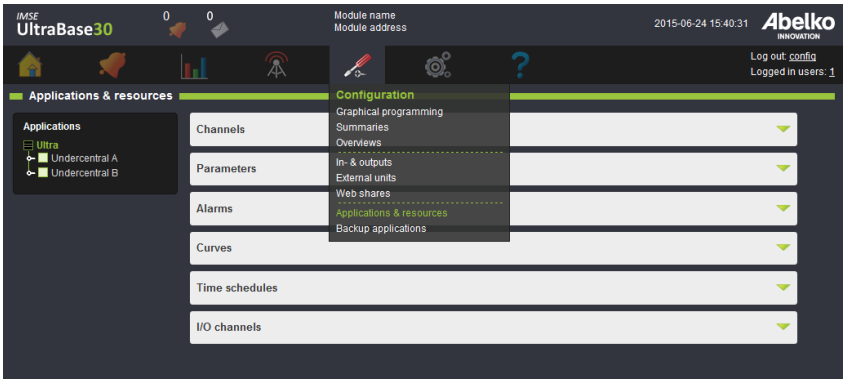
Summaries are menu pages with resources you added to see and edit, such as channels, parameters, alarms and more.

Overviews are pictures you create of an installation and add important values to quickly and easily see the status of the installation. Drag and drop



resources from the node tree on the left onto the workspace to add them. If you wish to use pictures for the overview, upload them by the page **System** and **File manager**.

When you have finished creating overviews and summaries, they will be available at the **home page**.



If you need to change settings later or get an overview of applications, resources, inputs and outputs, external units and more, go to the menus located under the graphical programming.

There are two buttons in the interface:

The bell indicates if there are any active alarms in the system, and the letter shows if there are any new notes. By clicking on these, you will be directed to more detailed information.

This was a short overview of the system. For more detailed information see the relevant chapter.

3 Applications & Resources

Under the menu **Configuration** and **Applications & resources** there is an overview of the system's parts and where you can edit them. In this chapter we will explain the different types of applications and resources and their features.

3.1 Applications

An **application** is a box which contains a composition of functions. It is in the application that you will decide which features all channels, alarms, parameters and more will have and how they will work together.

Applications are created in the graphical programming, which in turn creates the underlying script. A tool has been added to help you to create substation, ventilation and PID controllers.

3.2 Channels

Channels handle all the variable data in the unit. I/O channels are connected to physical inputs and outputs and have scale factors for sensors and actuators. Ordinary internal channels handle calculating values. Channels can be monitored by alarms, and their values can be logged.

3.3 Alarms

An **alarm** monitors the channel value according to set conditions and limits and is always created based on the channel. **A-alarm** has a red indicator and **B-alarm** has a yellow indicator.

In addition to the **A-** and **B-alarms**, you can assign an alarm a priority from **C-Z** (blue indication). You can also specify settings for acknowledgment and resetting an alarm.

Acknowledgment is a receipt that you have received and acknowledged the alarm by filling in your name. This is stored in the **Alarm history**, which is a list of all alarms and events.

3.4 Alarm Groups

Alarm groups simplify managing alarm sendouts and allow creation of features for multiple alarms at the same time. You can select individual alarms or alarms of a specific priority.

3.5 Parameters

A **parameter** is a value that is named and set by the user. When you use the controller tool, parameters will be automatically created, including set points that you can edit later.

3.6 Curves

A **curve** is a interpolation table presented as an configurable curve. Curves are used by controllers to do things like converting an outdoor temperature into a flow temperature and can be monitored by the user both graphically and numerically.

3.7 Time Schedule

A **time schedule** is used to set times when things should be active or inactive. You can add rules which can be repeated weekly, monthly, yearly or with a free adjustable interval. You can also choose not to repeat the rule and only set times for a specific date.

4 Graphical Programming

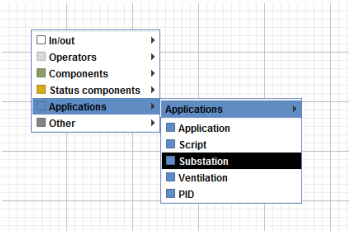
In this chapter you will learn how to use the graphical programming. At the end of this chapter there is an overview of all the icons and their functions.

The graphical programming is located under the **Configuration** menu.

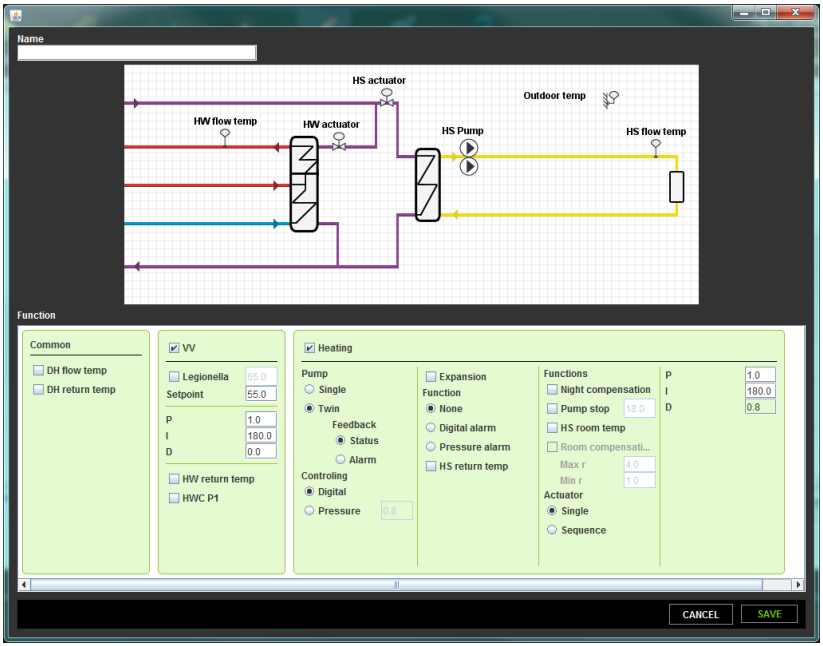
You work with different levels in the graphical programming. On the top level you create head applications, which can be started and stopped independently. It is here you connect the physical inputs and outputs. Inside of these head applications, you have sublevels where you create subapplications. It is here you add components.

4.1 Create Controllers

Create controllers is a tool in the graphical programming that helps you to create controllers quick and easy.



1. **Right-click** on the workspace and open menu **Applications**. Choose what type you want to create (**substation**, **ventilation** or **PID** controller).



2. Choose what functions you want to have in the system in the new window. Don't forget to **name** the controller.

3. Click **Save**. A graphical programming will be created where you can make adjustments. The controller tool will also generate an overview and a summary.

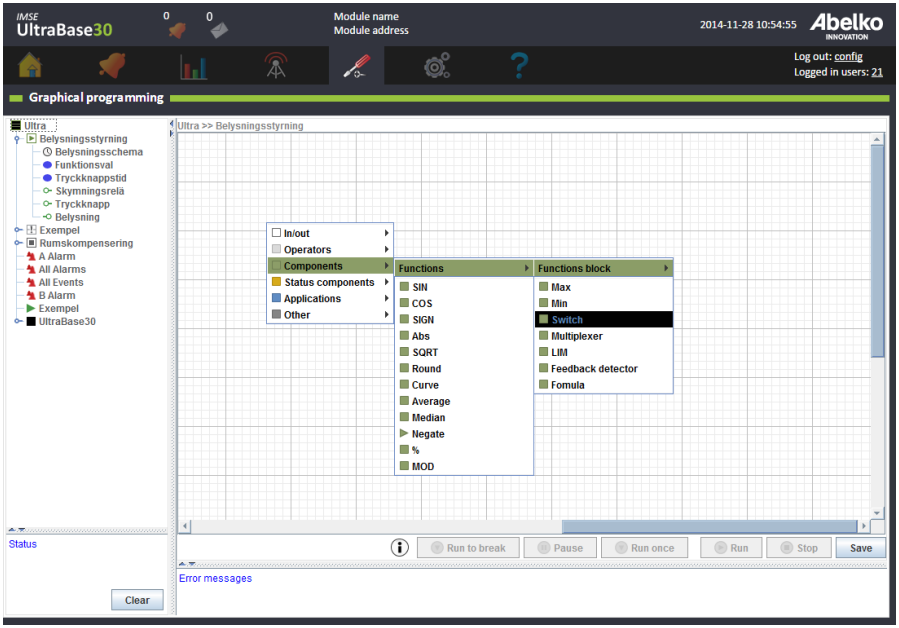
4. Verify the generated application is working properly.

4.2 Create Graphical Programs

In order to build a system you need to create applications. These will contain all the components, which in their turn creates a system with different functions.

1. To create an application **right-click** on the workspace and choose **Application** in the menu.

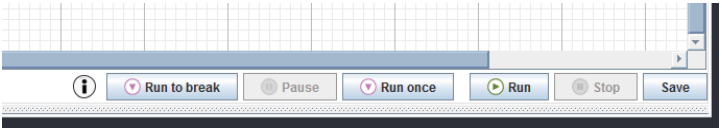
2. **Double-click** on an application to enter. In this menu you can create subapplications and add resources and components by **right-clicking** on the workspace.



3. **Drag-and-drop** channels, parameters and more from the node tree to the left onto the workspace to add them.

4. Connect the resources and components.

Note! Before an application is ready to be used, it needs to be saved and to be **runnable**. For it to be **runnable,** all the components in the application and all subapplications need to be connected to each other so it can create a syntactically correct script. The text box down to the left gives information about the script. If it is not **runnable**, the head application box will turn grey and have an exclamation mark.



5. You can use one of the debug states if you want to test a script for an application. Mark the application and use the buttons down to the right (**Run to breakpoint**, **Pause** or **Run once**). The head application will turn pink and a pink symbol will appear. Breakpoints in the script will be ignored when the application is not in the debug state.

6. To run the application, select it and press **Run** at the bottom right. It will turn green and a play icon will appear. When an application is running, you can only move resources and components. If you want to **edit** them or add new ones, you must first stop the application.

7. To go to settings or remove an application, **right-click** on the application.

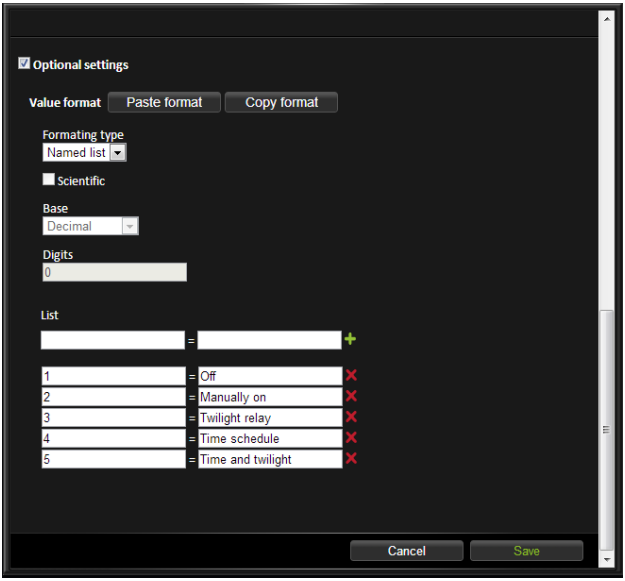
8. Do not forget to **save** at the bottom right of the page. The applications that you have made changes to will have a dashed frame so it is easy for you to see which ones that needs to be saved.

Example: create a lighting control

In this example, we want to control an outdoor lighting. We start with creating a new application and call it *Lighting control*.

With a parameter, we will be able to choose how it should be controlled. We create a parameter *Function selection*, and under optional settings choose *Named list*.

We create a list with:



1 = *Off*

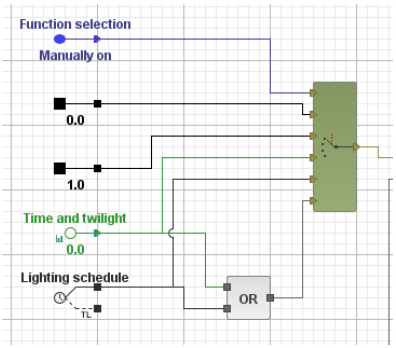
2 = Manually on

3 = Twilight relay

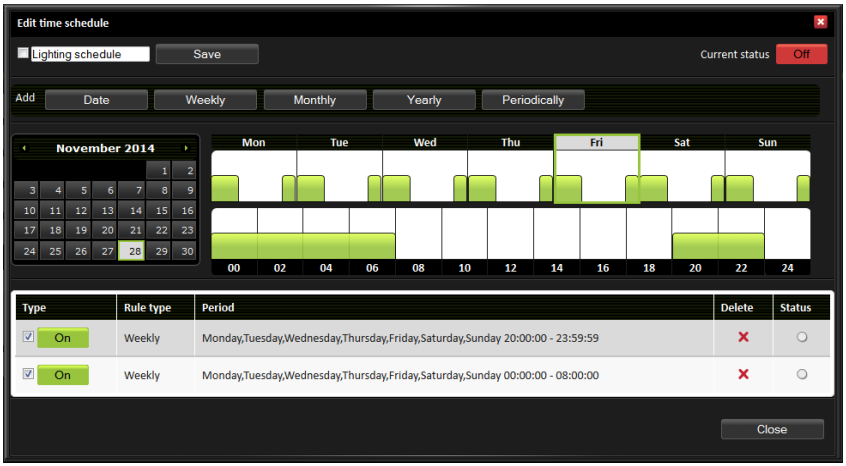
4 = *Time schedule*

5 = *Time and twilight*

With a multiplexer, that we pull out so that it has five inputs in addition to the top control input, we let the parameter select functions. The inputs are numbered with start from the top so that the first input corresponds manually off and should, therefore, be zero.



The next number should be for *Manually on*, and then we create an in channel for the twilight relay and create a time schedule.

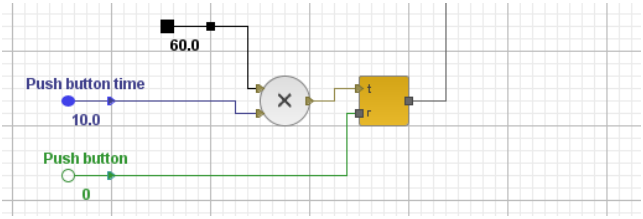


We create the last alternative *Time and twilight* by connecting the twilight relay with the time schedule with an OR-function.

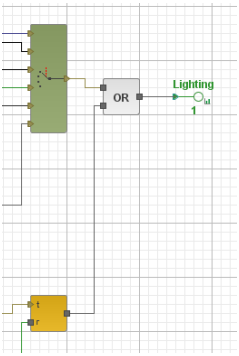
Another feature we want to add in the lighting control is a button that turns on the lights and then turns them off automatically after a predetermined time.

First, we create an in channel for the button and a parameter for the set time. If we choose to set the time in minutes, we need to multiply it by 60 to get seconds. We connect these two to a TIMER-component. The T-input states time and the R-input is a reset input. The components act as a stove

timer. When r is one, the timer is turned on, and the outcome is one. When r gets low, the timer is released and starts the countdown. When r becomes zero, the output becomes zero.

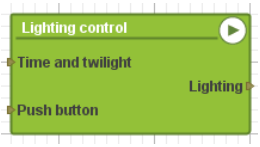


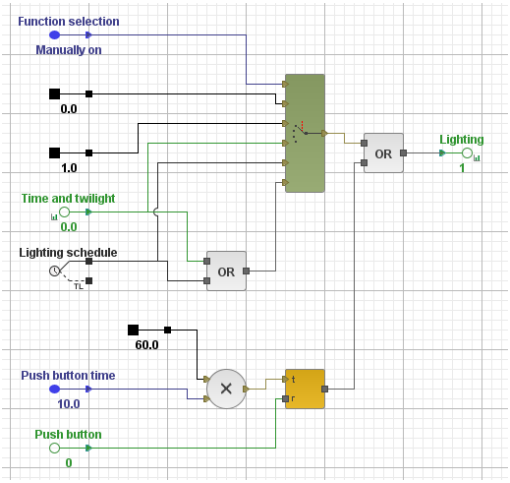
The multiplexers and the timers' outputs are connected to an OR-function and continue to an out channel that will to control the light.



The lighting control is now complete and ready to be used.

This is what the finished graphical programming and application looks like.

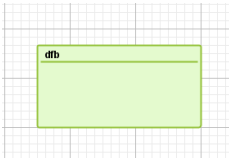




Application types

Applications (head applications, have a status symbol, located on the top level in the node tree).

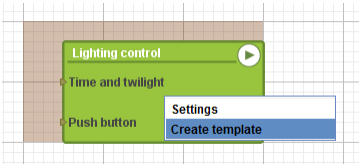
Subapplications (have no status, located under an application in the node tree).



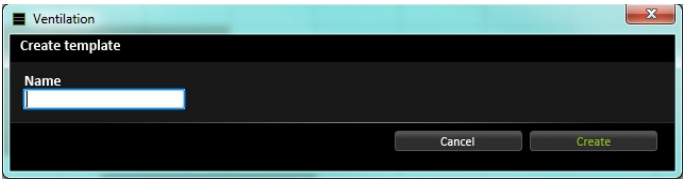
Script applications (marked with a notepad)

Locked applications (marked with a lock, these are not editable)

4.2.1 Create a Template of a Subapplication



1. **Right-click** on an application and choose **Create template**. A template is a snapshot of how that application looks at the time it was taken.

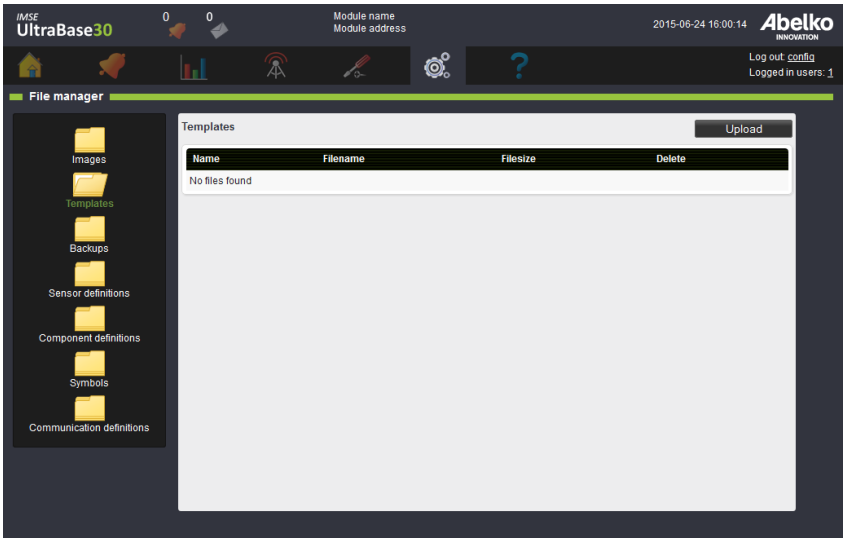


2. Name the template and add a description.

3. Click **Save**.

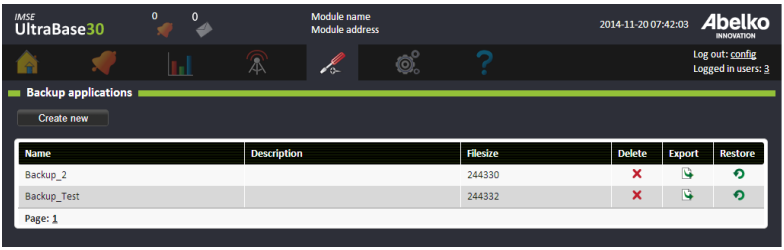
The template will be available when you create a new application. If you choose to use the template, the new application will be a copy of the old application including overviews and summaries.

If you want to change a template, you first need to create an application based on that template, make the changes and then save it as a new template.



To upload a template, go to the menu **System**, **File manager**. Select the folder **Templates** and upload the file. It will be available when you create a new application in the graphical programming.

4.2.2 Backup Applications



Backup applications can be found under the **Configuration** menu. These backups can be used to go back to a graphical programming that worked before you made changes without affecting logged data in the databases.

Create a new application backup

1. Click **Create new**.

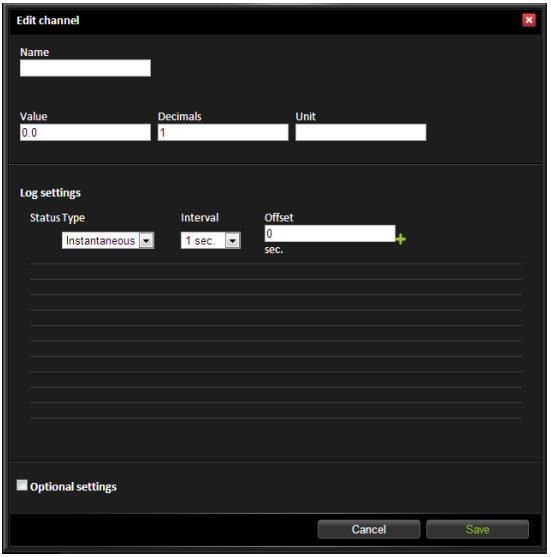
2. Name the backup and add a short description.

3. Click **Save**.

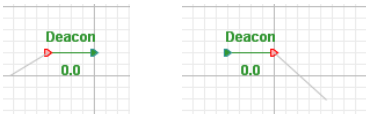
The backup is now saved in the system and placed in the list of existing backups.

4.3 Channels

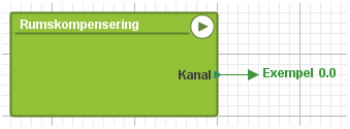
1. Create a channel by **right-clicking** on the workspace in the graphical programming and select **Channel** from the menu. Edit the channel and click **Save**. The channel will now be on the workspace.



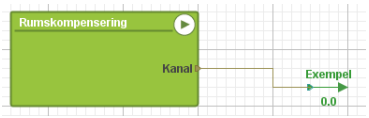
2. Connect a channel by dragging and dropping the arrow to the resource you want to connect it with. You can control how the value will be used with help by the arrows. The left arrow will connects in to the channel, and the right arrow connects out from the channel. You cannot use both from the same channel.



When you connect channels with an application, the channels will merge with the application in order to make room for more channels.



By moving a channel away from the application, it will return to normal mode so that you can connect the channel with more than one resource.



Channel types, top level

Analog inputs/outputs: these are representations of the analog inputs and outputs available on the device. These can handle a rescaling of the measurement unit. For the **linear sensor type**, the rescaling will have space for scale factor and offset, but also measuring range with minimum and maximum values. For the **non-linear sensor type**, you need to fill out multiple values to get a curve. The more values you add, the more fine-tuned the curve will be.

Digital inputs/outputs: these are representations of the digital inputs and outputs available on the device. Digital inputs can be used several times and measure status and frequency and act as a counter. Digital outputs can be inverted. This can be done under the settings **Scaling** by clicking a checkbox.

Universal inputs/outputs: these are representations of the universal inputs and outputs on the device. Universal inputs can be configured to measure resistance, voltage, current or be used as a digital input status. You can read more about its settings in the chapter **Inputs & Outputs**.

Channel types, sublevel

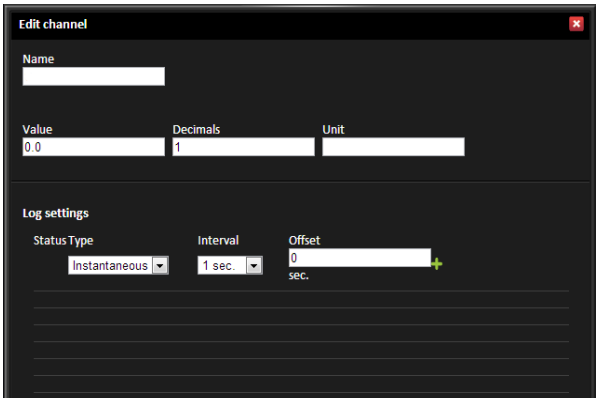
Channel: handles variable information and can modify information by applying scale factors and other mathematical functions.

In channel: when you create an in channel in a subapplication, you create an input on the head application.

Out channel: when you create a out channel in a subapplication, you create an output on the head application.

4.3.1 Channel logs

1. **Right-click** on a channel and select **Settings**.



2. Choose **type** of log, set **range** and **offset**.

3. Click on the **plus** to add a log. You can have different types of logs for the same channel.

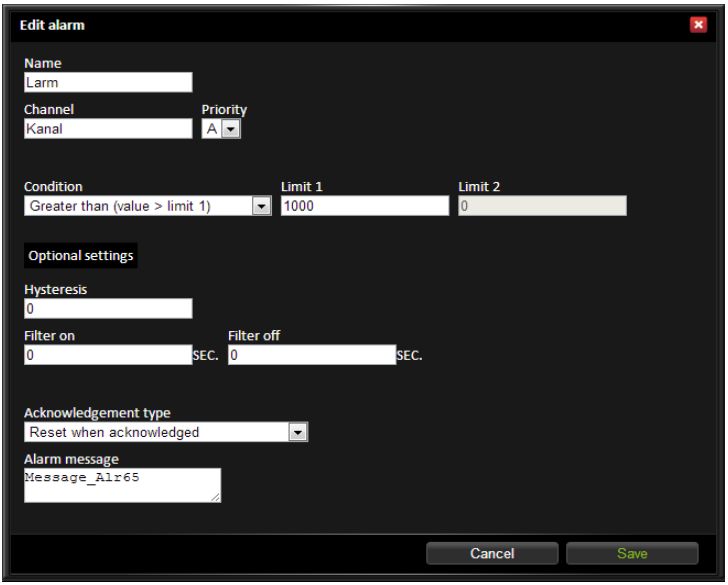
4. Click **Save**.

There are two types of logs. **Smart log** is a log that compresses data as it comes in. Second values are converted to minute values, and so on. This allows one to store data for a longer period. The other type of logging is periodic, where you set the log's time interval. You can log momentary value, average value, maximum value, minimum value or sum for a certain period.

See more about logs in the chapter **Data Logs**.

4.4 Alarms

1. To create a new alarm in the graphical programming, **right click** on a channel in the node tree or in the workspace and select **Alarm**.



2. Under settings, you can change the name of the alarm, see which channel it belongs to, and change the priority and alarm limits.

   The different **priorities** are:

- **A-alarm** (red indication)
- **B-alarm** (yellow indication)
- In addition to these, an alarm can have the priority **C-Z** (blue indicator)

The different **conditions** are:

- **Equals (value = limit 1)**: the alarm is activated when the value is exactly the same as the value entered under Limit 1.
- **Greater than (value > limit 1)**: the alarm is activated when the value is greater than the value entered under Limit 1.
- **Less than (value < limit 1)**: the alarm is activated when the value is less than the value entered under Limit 1.
- **Greater (abs (value) > limit 1)**: the alarm is activated when the value is greater than the value entered under Limit 1 without taking into account whether the value is positive or negative.
- **Smaller (abs (value) < limit 1)**: the alarm is activated when the value is smaller than the value entered under Limit 1 without taking into account whether the value is positive or negative.
- **Between (limit 1 < value < limit 2)**: the alarm is activated when the value is between the values entered under Limit 1 and Limit 2.
- **Outside (limit 1 < value or value > limit 2)**: the alarm is activated when the value is below the value entered under limit 1 or above the value under Limit 2.

The different **acknowledgments** are:

- **Reset when acknowledged**: the alarm remains active until a user acknowledges it. If the error remains after the alarm is acknowledged, it will remain active until the error is cleared, including the delay and hysteresis.
- **Automatic reset, with acknowledgment**: the alarm reset itself when the error is cleared, but it still has to be acknowledged. In this case, the acknowledgment is a kind of confirmation that someone has noticed the alarm. Details of the reset and the acknowledgment are both recorded in the alarm history.
- **Automatic reset, without acknowledgment**: there is no need for a user to acknowledge the alarm. The alarm is cleared when the error is no longer active, as defined in the alarm conditions. You can specify delay

and hysteresis values in the condition.

4. You can edit **Hysteresis** and **Filter** under **Optional settings**.

The **Hysteresis** setting prevents the alarm repeatedly switching between active and inactive when the value is equal to the limit. It specifies the number of degrees by which the value must fall back below the limit or limits before the alarm can be cleared.

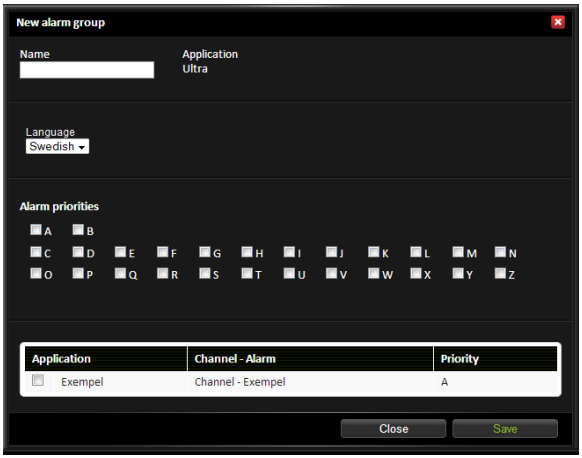
The **On-filter** specifies the number of consecutive seconds the alarm condition must be true before the alarm is triggered. The **Off-filter** specifies the number of consecutive seconds the alarm condition must be false before the alarm is cleared.

4.5 Alarm Groups

Alarm groups simplify managing alarm sendouts and allow creation of features for multiple alarms at the same time. For example, in the graphical programming you can use an alarm group that contains all **A-alarms** instead of working with all of the **A-alarms** one by one.

1. **Right-click** on the workspace when you are inside of an application; select menu **In/out** and choose **Alarm group**.

2. Name the alarm group.



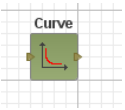
3. Select which alarms you want in the alarm group by clicking the box next to the alarms. You can only choose alarms from the application and its subapplication that you have selected.

4. For example, if you would like to add all **A-alarms** from that application, click the box next to the **A**. By doing this, all future **A-alarms** created in the application will be included in the alarm group.

5. Click **Save**.

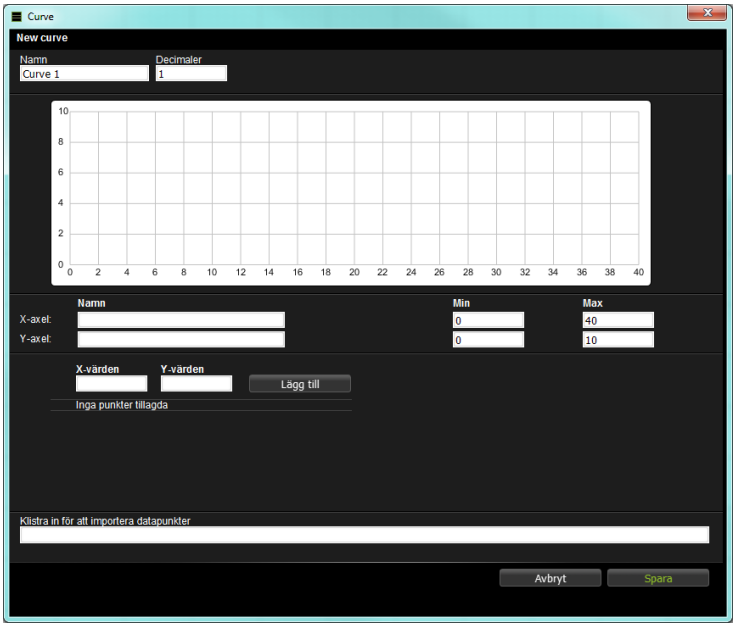
Text and email settings for sendouts are configured under the menu **Data**, **Alarm sendouts**.

4.6 Curves



Curves are used to convert or compensate values.

1. Start by **naming** the curve.



2. Select the **number of decimals** you want the data points to have.

3. Choose **min** and **max** values for X- and Y-axis.

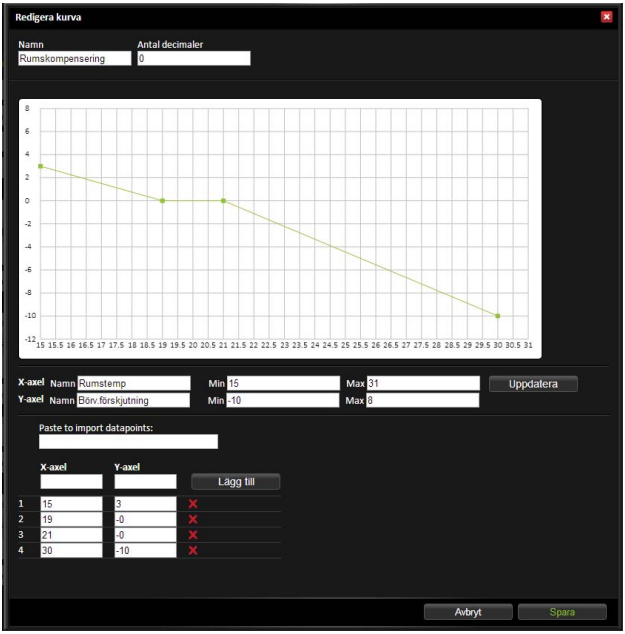
4. You can now start to add data points by **double-clicking in the graph** and adjust them by moving them around. You can erase a point by **double-clicking** on it.

5. You can also add and edit data points in the table under the graph. If you have a text file, you can paste it in the window **Paste to import data points**.

6. Click **Save**.

Example: a curve for room compensation

If you want to have a compensation from the room temperature in a controller, creating a curve is a good and easy way to set the compensation level.



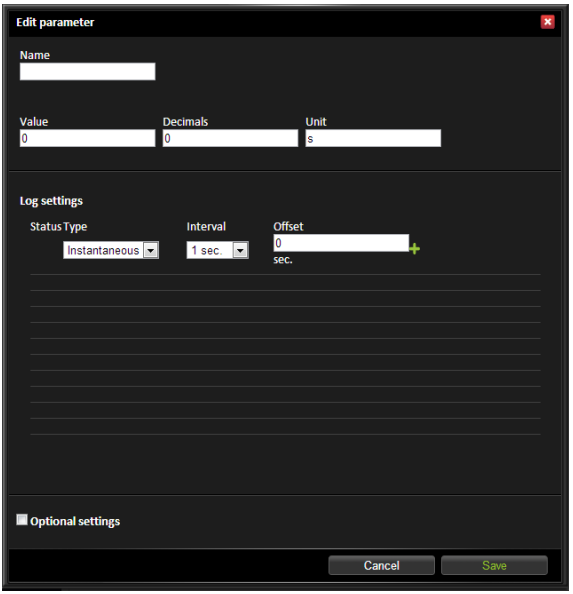
The picture shows a curve with +3K compensation at 15°C with no compensation between 19-21°C and -10K at 30°C.

Connecting the curve to a room sensor is done in the graphical programming.

4.7 Parameters

Parameters are used to store values that can be changed by a user. You use these to set values, which in their turn control other parts of the plant.

1. **Right-click** on the workspace when inside an application; select menu **In/Out** and click **Parameter**.



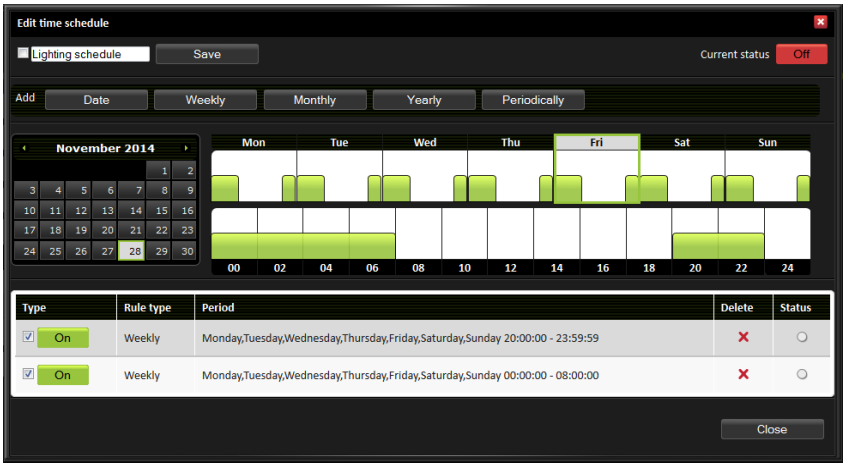
2. Name the parameter. Enter a value, number of decimals and a unit. You can choose if you want to log the parameter. Add a new log to the list by clicking on the **plus**.

3. You can change the **value format** under **Optional settings**. For example, this can be used to convert C° to F.

4.8 Time Schedules

A **time schedule** is used to set times when things should be active or inactive. You can create a time schedule and connect it to an object in the graphical programming. You can also choose if you want the time schedule to enable or disable an object (the red blocks always rule over the green blocks).

1. **Right-click** on the workspace when you are inside an application; select menu **In/Out** and click **Time schedule**.

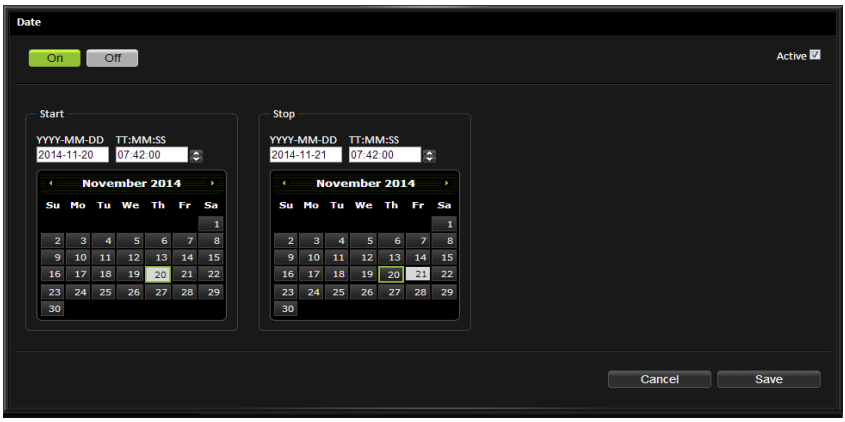


2. Choose which type of rule you want to add: **Date**, **Weekly**, **Monthly**, **Yearly** or **Periodically**. We will explain the different types in the next section.

3. When you added rules, you can see the time schedule's **status** on the top of the page. To **disable** a schedule uncheck the box to the left of its name. The time schedule will not become active until you select the box again.

Date

This is used when you want to do something on a specific date.



1. Select if you want the rule to enable (**On**) or disable (**Off**) objects.

2. Set start and stop time in the boxes **YYYY-MM-DD** and **TT:MM:SS**.

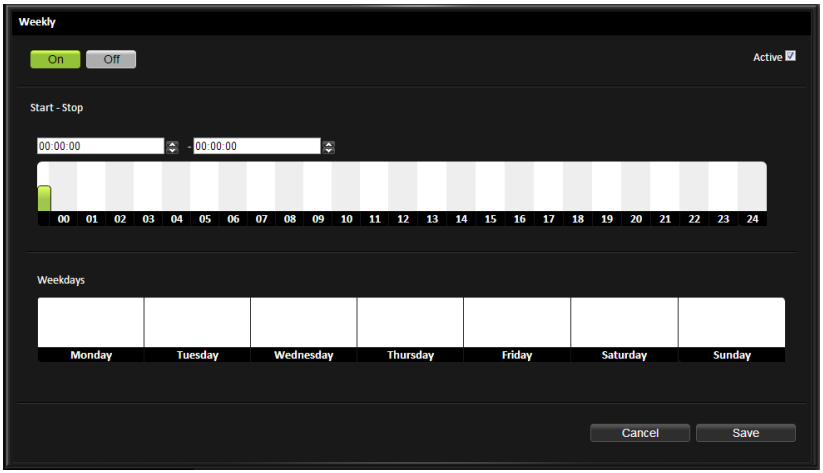
3. Mark the date in the calendar.

The rule will automatically be set to **Active**. You can **Inactivate** it by using the checkbox up in the corner. If you click **Cancel**, you will leave you the page without saving the changes you have made.

4. Click **Save**. The rule will now be included in the list on the overview page.

Weekly

This is used when you want something to be repeated every week.



1. Select if you want the rule to enable (**On**) or disable (**Off**) objects.

2. Set start and stop time by moving the bar in the time axis or type the time in the boxes.

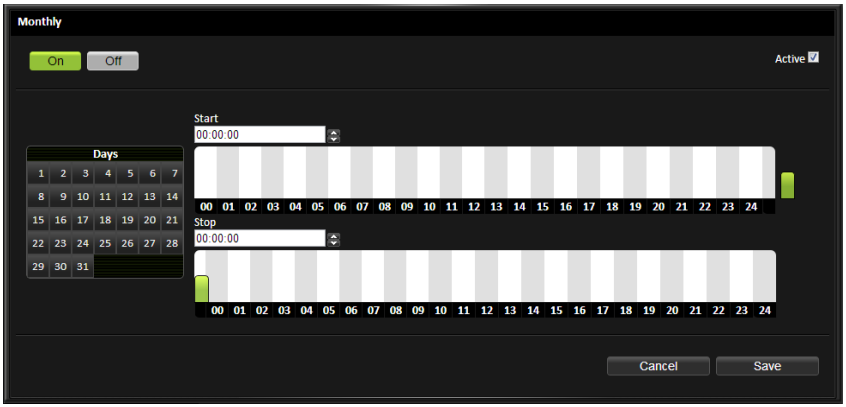
3. Select what days the rule should apply to by clicking on the weekdays.

The rule will automatically be set to **Active**. You can **Inactivate** it by using the checkbox up in the corner. If you click **Cancel**, you will leave you the page without saving the changes you have made.

4. Click **Save**. The rule will now be included in the list on the overview page.

Monthly

This is used when you want something to be repeated every month.



1. Select if you want the rule to enable (**On**) or disable (**Off**) objects.

2. Set start and stop time by moving the bar in the time axis or type the time in the boxes.

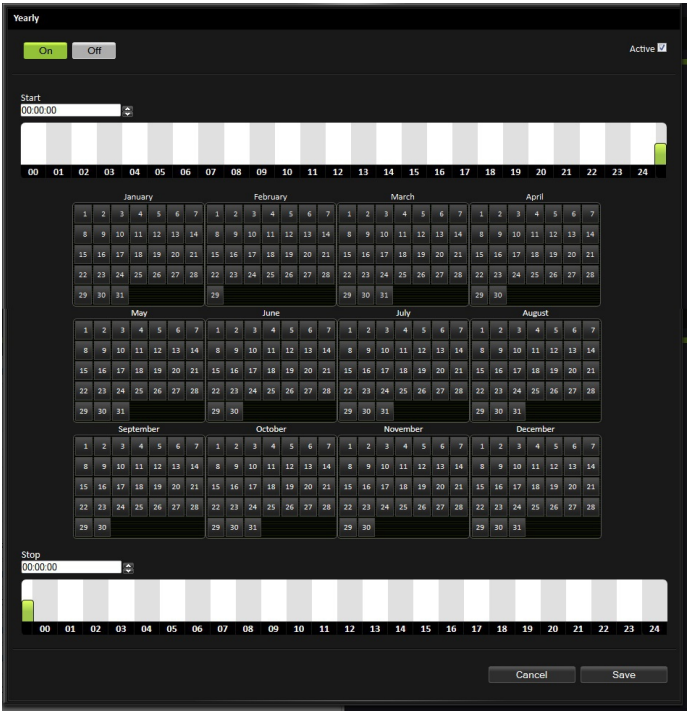
3. Mark the date in the calendar.

The rule will automatically be set to **Active**. You can **Inactivate** it by using the checkbox up in the corner. If you click **Cancel**, you will leave you the page without saving the changes you have made.

4. Click **Save**. The rule will now be included in the list on the overview page.

Yearly

This is used when you want something to be repeated yearly.



1. Select if you want the rule to enable (**On**) or disable (**Off**) objects.

2. Set start and stop time by moving the bar in the time axis or type the time in the boxes.

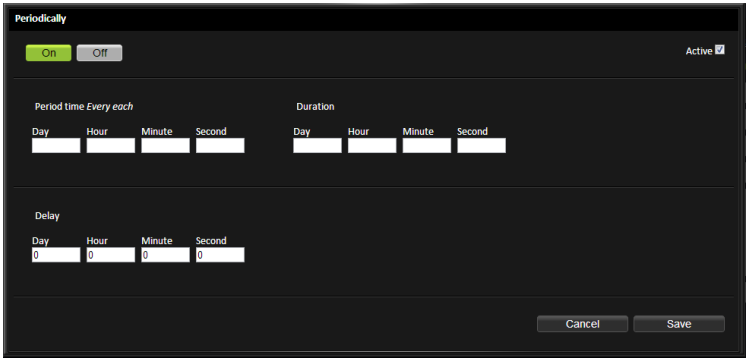
3. Mark the date in the calendar.

The rule will automatically be set to **Active**. You can **Inactivate** it by using the checkbox up in the corner. If you click **Cancel**, you will leave you the page without saving the changes you have made.

4. Click **Save**. The rule will now be included in the list on the overview page.

Periodically

This is used when you want something to be repeated a certain period.



1. Select if you want the rule to enable (**On**) or disable (**Off**) objects.

2. Set **Period time**, this will decide how often the period should be repeated. For example, if you want something to be repeated every third day, type 3 under **Day**.

3. Set for how long you want it to be active under **Duration**. For example, two hours by typing 2 under **Hour**.

4. You can also add a **Delay**. This gives a delay into the period when the duration will be active. For example, five hours by typing 5 under **Hours**.

This means that it will be active every third day during 05:00 and 07:00.

The rule will automatically be set to **Active**. You can **Inactivate** it by using the checkbox up in the corner. If you click **Cancel**, you will leave you the page without saving the changes you have made.

5. Click **Save**. The rule will now be included in the list on the overview page.

4.9 Applications Status

Applications can have different statuses and they are indicated both on the application and in the node tree. Subapplications always have the same status as the head application.

Run: is used to run the application normally.

Stop: is used to stop the system, for example during maintenance. No script is running, all the alarms are blocked and channels will go back to their initial values.

In addition to these statuses, you can put the system in **debug mode** if you want to test the script. To use this mode, you must first stop the application by selecting the application and pressing **Stop** at the bottom of the page.

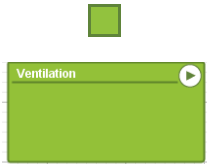
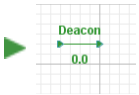
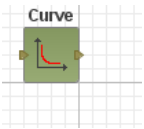
Pause: can be used for minor maintenance since it does not affect channel settings. Script execution is stopped, variables are not reset and there is no re-initialization of channels. Alarms and logs will function as normal. When the **run once** and **run to break** has been run through, they will finish in **Pause**.

- **Run once**: is used to run the system to end or break point. When it has run once it will end in the paused mode.

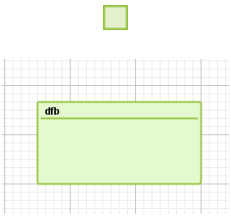
- **Run to break**: is used to run the system to a break statement in a script. When it has run to a break piont it will end in the pause mode. Break points can only be used in script; you can read more about this in the Reference Manual (available at abelko.se).

You can also read about the different statuses in the web interface by clicking the **i** icon next to the status buttons.

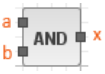
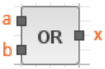
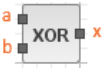
4.10 Overview Icons Graphical Programming

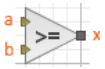
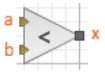
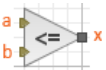
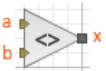
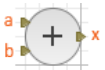
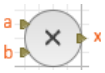
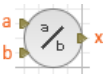
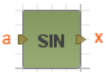
	Application (in the node tree and on the workspace)
	Error-status
	Debug-status
	Channel
	Curve
	Run-status
	Alarm
	Alarm group

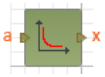
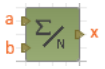
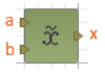
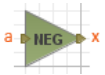
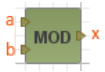
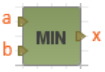
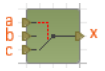
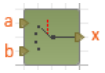
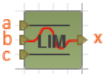
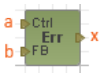
	Log
	Locked application
	Parameter
	Pause-status
	Summary
	Script application
	Stop-status
	Time schedule
	UltraBase
	UltraTouch

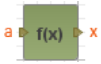
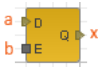
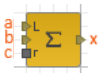
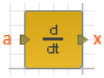
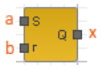
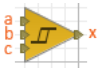
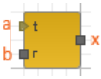
	Subapplication (in the node tree and on the workspace)
	URL
	Overview

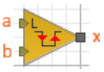
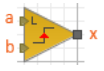
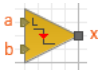
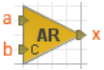
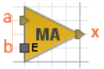
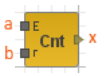
4.11 Explanation of the Different Components

Operatorer	
	ANDNAND if both A and B are true (non-zero), the output (x) is a. In all cases where x is one, y is one.
	**AND** if both A and B are true (non-zero), the output (x) is a.
	**OR** if either a or b or both are true (non-zero), the output (x) is a.
	**XOR** if a or b, but not both, are true (non-zero), the output (x) is a.
	NOT if a is true (nonzero), the output x is zero. Otherwise, x is one.
	Equal if A and B are equal, the output (x) is a.
	Greather than if a is greater than b, the output signal (x)is a.

	**Greater equal** if a is greater or equal to B, the output (x) is a.
	**Less than** if a is less than b, the output signal (x) is a.
	**Smaller equal** if a is smaller or equal to b, the output (x) is a.
	**Not equal to** if a is different from b, the output signal (x) is a.
	**Plus** a plus b is the value of the output signal x.
	Minus a minus b is the value of the output signal x.
	**Multiplication** a multiplied by b, is the value of the output signal x.
	**Division** a divided by b, is the value of the output signal x.
Komponenter	
	**SIN** x = sin(a) where a is an angle in radians.
	COS x = cos (a) where a is an angle in radians.
	SIGN keeps track on which value the input signal a has. If a is less than zero, the output signal x = -1. If a is greater than zero, x = 1. If a is exactly zero then x is also zero.
	Absolute value x = abs(a), i.e. that a possible minus sign is removed.
	SQRT the square root of a is the value of the output signal x.
	Round rounds the input signal a to the nearest integer which is the value of the output signal x.

	**Curve** output x is the value at the y-axis corresponding to the input signals (a) position on the x axis.
	**Mean** is calculated on the input signals, the value of the output signal is x.
	**Median** is calculated on the input signals that will be the value of x.
	**Negating** x = a * -1
	% a and c are set values that are compared with b. The output x is a percentage where between a and c b is. x = 100 (b-c) / (a-c), limited to be between 0 and 100. Useful for sequence control.
	**MOD** (Modulus) x is the remainings after the division a / b.
	Max choose the larger of the input signals as value for output signal x.
	**Min** choose the smallest of the input signals as value for output signal x.
	**Switch** if a is zero, the output signal x equals c. If a is true (non-zero), the output signal x equals b.
	**Multiplexer** the input signal a select which of the other input signals will be output signal. If a = 0 then x = b, if a = 1 then x = c, etc.
	**LIM** is a component that limits a signal. A specifies the upper limit, and c specifies the lower limit. If the input signal b is between the limits, the output x will have this value. But, if the input signal falls outside a or c, the output x will have the limit as a value.
	**Feedback detector** if a or b, but not both, are true (non-zero), the output (x) is a.

	**Formula** the input signal a is calculated from the specified formula, which will give the value of the output signal x.
Status components	
	**Latch** if the input signal b is true (non-zero), the output signal x is equal to the input signal a. If b is zero, the component will keep the previous value of the output signal. The signal will then freeze.
	**Sum** every second, the input signal is added to the sum, which is the value of the output signal x. If the absolute value of the sum is greater than a (Limit), the value is not increased. When the input signal c (RESET) is different from zero, the sum is set to zero.
	**Derivata** output signal x is how much the input signal a has changed since previous second.
	**RS flip flop** if the input signal a (set) is true (nonzero), the output signal x has the value one. X remains one until b (reset) becomes true, then x is set to zero. At start-up, x is zero.
	Date x is the current year, y is the current month (1-12) and z is the current day of the month (1-31).
	Time x is the current hour (0-23), y is the current minute (0-59) and z is the current second (0-59).
	**Hysteresis** a and c are limit values. When b is higher than a, x is set to one. When b is less than c, x is set to zero.
	**Timer** output signal x is one, during the number of seconds that a (timer) specifies. Then it becomes zero. If b (reset) reaches zero, the clock will start again and x is one for a seconds. At startup, the clock is active and x is one.

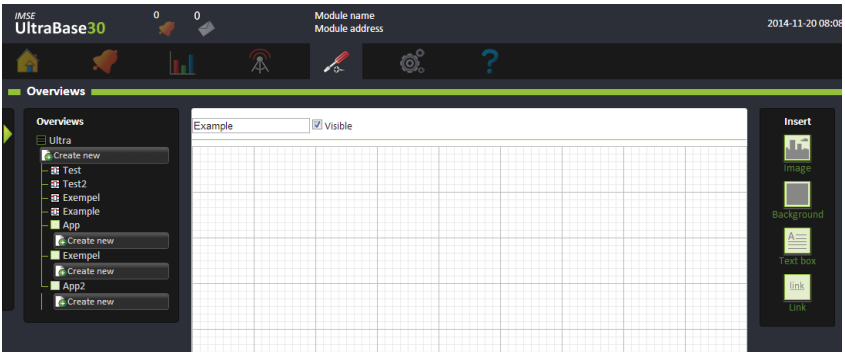
	**Flank detector** output signal x becomes one each time the input signal b passes the limit value a.
	**Flank detector pos** output signal x becomes one each time the input signal b passes limit value a on the way up.
	**Flank detector neg** output signal x becomes one each time the input signal b passes limit value a on the way down.
	**AR filter** autoregressive filter with filter factor b (Coefficient), which should be between zero and one. Zero means no filtering and one that the input signal a never goes through to the output signal x.
	On/Off delay delays the input signal b. When b is set to true, it must be true a seconds in a row before the output signal x becomes true. When b is set to zero and x is one, the x is changed to zero when b been zero c seconds in a row.
	Shift register with adjustable length (N). The input signal a is delayed for N seconds in the component, before it comes out in x.
	**MA-filter** (Moving Avarage-filter) the filter has a buffer for values. The number of values that can fit in the buffer is adjustable. Every update (second) which b (Enable) is true, the value from a is stored in a buffer and the oldest value is lost. The output signal x is the average of the values in the buffer.
	**Counter** if a (Enable) is true, then x is added up every seconds. If b (Reset) is true, then x is set to zero.

5 Overviews

Overviews provide a quick overview of the plant. You can add present values, alarms and more to this picture. In this chapter, you will learn how to create and edit overviews.

Overviews are created under the menus **Configuration** and **Overviews**. If you have used the controller tool in the graphical programming, there will already be a foundation for the overview. This is listed to the left with the same name as when you created them in the controller tool.

An overview created with the controller tool has a size of 720x300 pixels.



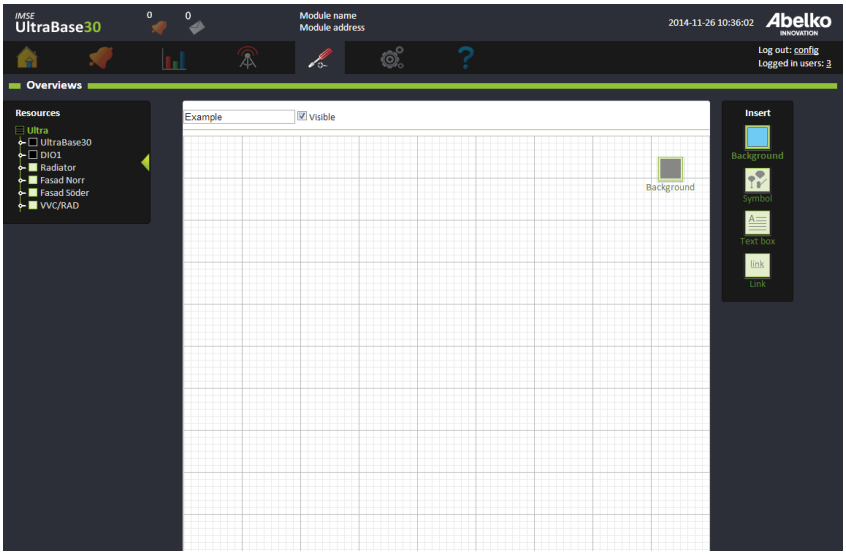
5.1 Create a New Overview

1. Decide for which application you want to create the overview. If it is not going to part of an application, you can create it on the top level (directly under the Ultra). If not, create it under the application you want to add values from. You can add and use channels, parameters and more from applications on the same level, as well as their subapplications.

2. Click **Create new** and name the overview.

3. Click **Save**. Now you can drag-and-drop resources from the node tree and edit them with the tools on the right.

5.1.1 Add Backgrounds



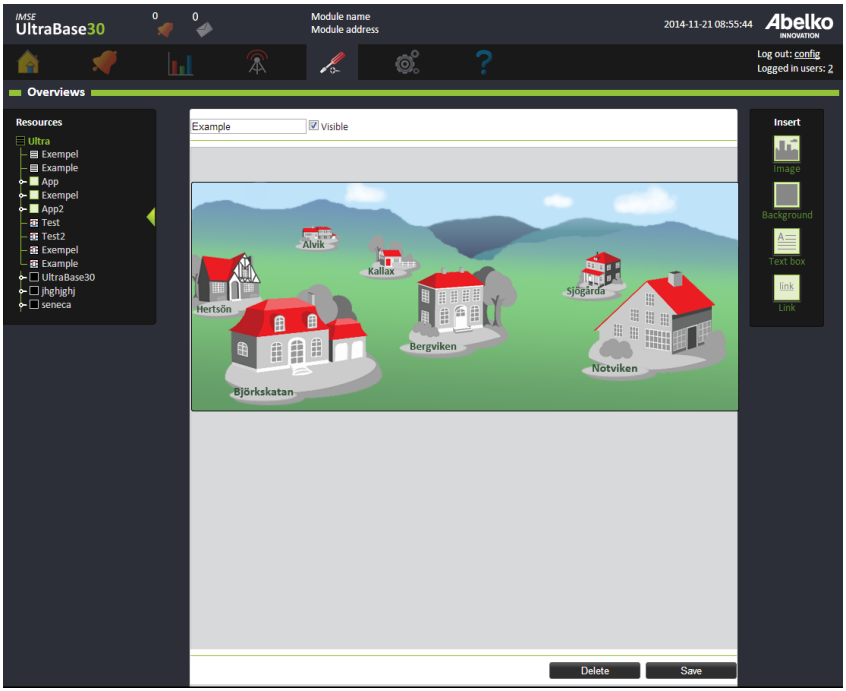
1. To add a background, use the **Background** tool in the **Insert** menu to the right. Grab the **Background** and drop it on the workspace.

2. You can select which image you want to use as a background, or you can also use a single color as a background. Once your background is determined, you may repeat it across the entire background or just parts of the background.

3. Click **OK**.

If you have an image as a background that does not cover the entire workspace, you can add a color as a background under the image.

5.1.2 Add Resources



1. When the image is in place, you can drag-and-drop resources from the node tree onto the workspace. For example, time schedules can be helpful, so you add it to the overview for quick access.

2. By hovering your mouse over an item, you can edit **view alternatives** or **erase** it. When you mark a value on the overview, it will be marked in the node tree. This also works in the opposite direction: if you mark

a value in the node tree, it will be marked out on the workspace.

Format is a view alternative where you can decide between viewing the value with unit and name or only viewing its value. By dragging-and-dropping the same resource and changing their view alternatives, you can split the resource to adapt it to the background.

A channel can also be viewed as an **Active meter**. This is a graph that will update when new values come in.

Symbol is a view alternative links a value with an image. Some of the images will change their appearance when the value changes (for example when a pump goes from inactive to active).

5.1.3 Add Text Boxes and Links

1. You can add your own text and links in the overview using the tools **Text box** and **Link**. These are located under the menu **Insert**.

2. You can easily link to other **overviews** and **summaries** by dragging-and-dropping them from the list. When you create links, you can link back to the other overview to easily navigate between them.

3. When you are satisfied with the overview, click **Save**.

The overview is now available in the node tree, and you can choose to have it as your homepage (this setting is located under the menu **System** and **Settings**). If you don't want it to be available to view, deselect the **Visible** box next to its name.

5.2 Tools

Background used to add a single colored background or an image as a background.

Symbol used to add symbols and images uploaded to the Ultra.

Text box used to add a box with text.

Link used to add a link to web pages and more. You can choose your own link text.

5.3 View Alternatives

Format shows the value like a text with different alternatives for name, value and unit.

Active meter shows the value as a graph, which is updated as new values come in. You can set the x-axis so it will remember and keep the max height in the graph.

Symbol shows the value as a symbol. Some of the symbols will change appearance when the value changes, for example when a pump goes from active to inactive.

6 Summaries

Summaries are views with selected channels, parameters, curves and databases. Here you can see their values and change settings. In this chapter, you will learn how to create and edit summaries. We start by going through an example and then explain the different tools and components in more detail.

You create summaries under the menu **Configuration**, **Summaries**. If you have used the controller tool in the graphical programming, there will already be a foundation for the summary. These are listed to the left with the same name as when you created them in the controller tool.

6.1 Create a New Summary

1. Deciding for which application you want to create the summary. If it is not going to part of an application, you can create it on the top level (directly under the Ultra). If not, create it under the application you want to add values from. You can add and use channels, parameters and more from applications on the same level, as well as their subapplications.

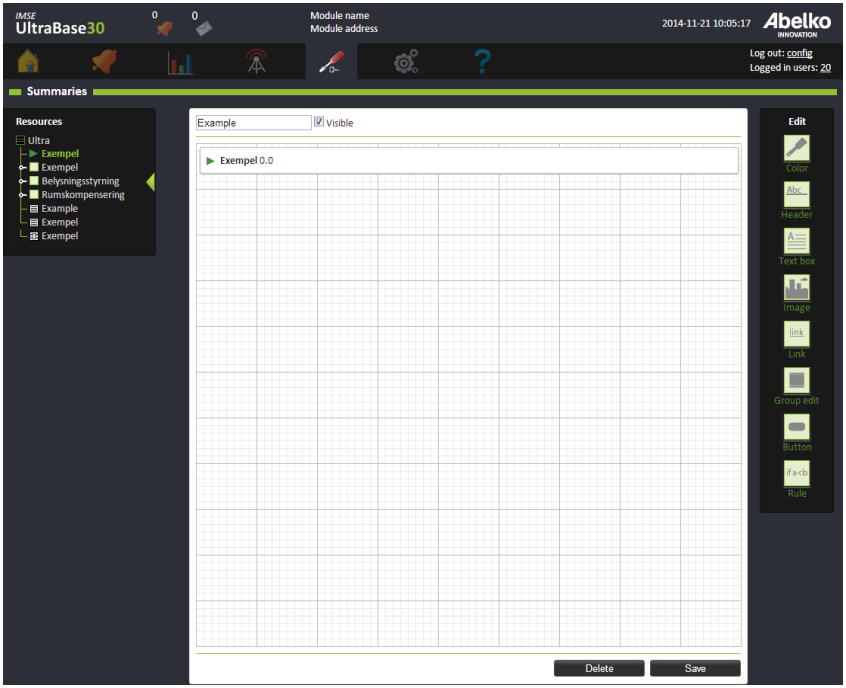
2. Click **Create new**, and name the summary.

3. Click **Save**. Now you can drag-and-drop resources from the node tree and edit them with the tools on the right.

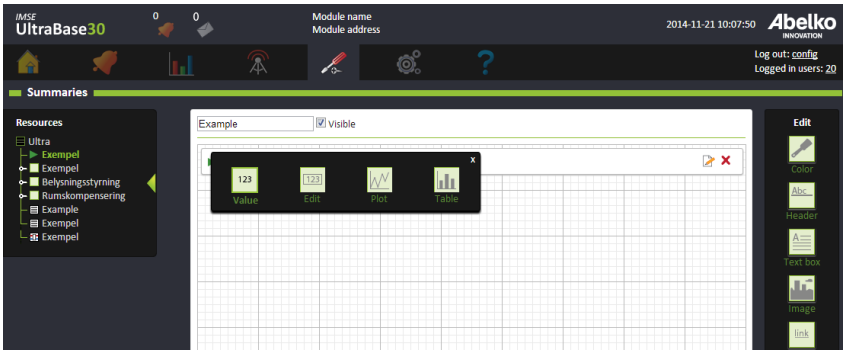
4. The tree view will extend when you click save. Add resources by dragging-and-dropping them on the workspace. Channels, alarms, set points, and more will be available under each application in the tree view. Time schedules are good to have, makes it easier to access and

edit them.

In the following figure we have added a channel. The default view shows name, value and unit.

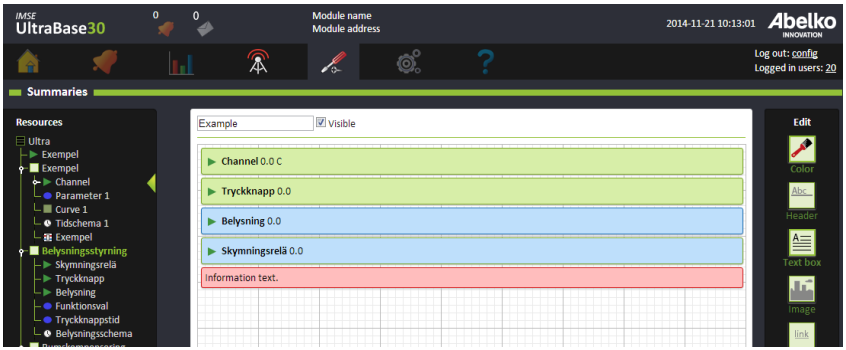


By hovering your mouse over an item, you can edit **view alternatives** or **erase** it.



5. To modify the appearance of the summary, use the menu **Edit** to the right. Drag the desired editing icon from the menu and drop it on the resource you wish to change.

   The following are a list of changes you can make with the **Edit** menu. **Color** changes the color of a frame, a text or a background. **Header** adds a break with a header; this helps to divide and group the resources. You can create **Text boxes** and add **Links** and **Images** that will be handled as an element in the list.



   Group edit edits multiple resources at the same time. **Button** sets several values at the same time. Drag-and-drop the resources you want into the selected area. The value you enter in the box will be the value that is copied to the other resources when you press the button. **Rule** controls the view of a row. For example, you can choose a row to be visible if an alarm becomes active.

6. To reorder the summary rows, drag-and-drop them into the desired order.

7. When you are satisfied with the summary, click **Save**.

It will now be visible in the node tree. To make it your homepage, open the menu **System**, **Settings**. If you don't want it to be visible, deselect the

Visible box next to its name.

6.2 Summary Tools

Color used to color backgrounds, frames and texts.

Header used to group and separate resources to organize overview.

Text box used to add a box with text.

Image used to add images uploaded to the Ultra.

Link used to link other overviews and summaries. You can choose your own link text.

Group edit allows you to group resources to edit them at the same time.

Button sets several values at the same time.

Rule adds rules for when something is shown or not.

6.3 Summary View Alternative

These options are accessed by hovering your mouse over a resource and clicking **Edit**.

Value shows the value like a text with different alternatives for name, value and unit. This is being used for static parameters like temperatures.

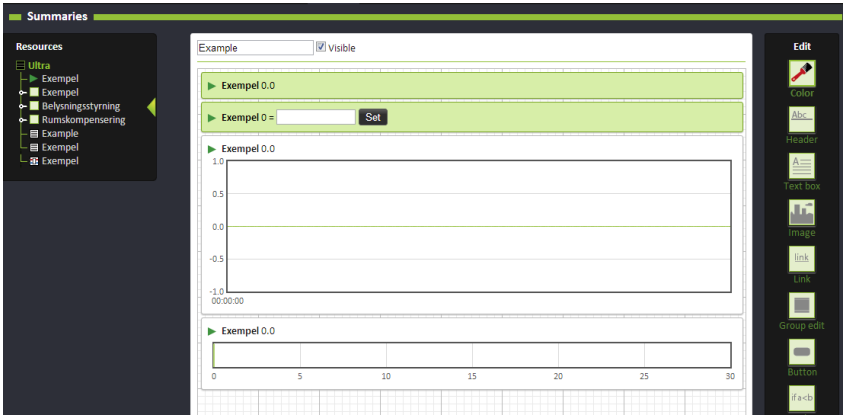
Editable allows manually editing the value of a resource.

Plot shows the value as a graph and is updated as new values come in.

Table is a bar chart that is automatically updated as new values come in.

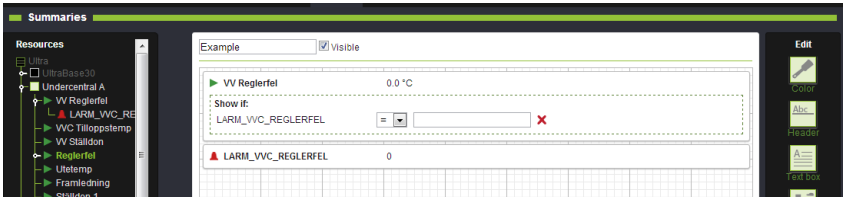
6.4 Channels

Channels can be viewed in several ways. As a default, it will appear in a row with name, value and unit. It can also be viewed as a plot or table; hover you mouse above it and click on the **Edit** icon.



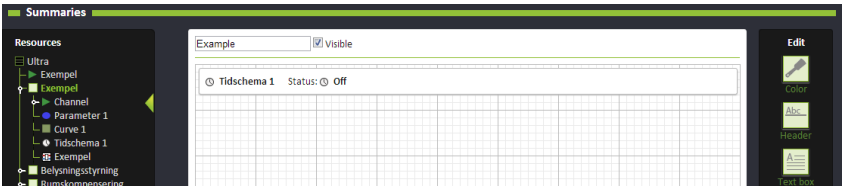
6.5 Alarms

When you add an alarm on the workspace, its name and current status will appear.



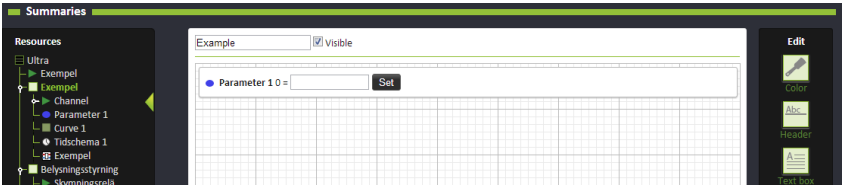
6.6 Time Schedules

When you add a time schedule from the node tree to the workspace, its name and current status will appear.



6.7 Parameters

Parameters can be both editable and non-editable. Hover the mouse over the parameter and click on the **edit** icon. Here you can choose if you want it to be editable (**Value**) or non-editable (**Set**).



6.8 Curves

To view curves as a graph, drag-and-drop the curve onto the workspace. When you have a finished summary, you will be able to edit the data points of the curve.

7 Alarms

This chapter will explain the different types of alarms. You will learn how to create and edit alarms and alarm groups. You will use these alarm groups to create alarm sendouts and black boxes.

You will find **Active alarms** and **Alarm history** under the **Alarms** menu.

7.1 Active Alarms

All active alarms in the system are listed on **Active alarms**. The number of active alarms will also be indicated on the top of the web interface. When it has a red symbol, it means that at least one of the active alarms is an **A-alarm**. When it has a yellow symbol, at least one of the active alarms is an **B-alarm**, and a blue symbol is when at least one alarm is a **C-Z alarm**. If the symbol is blinking, it means that there are alarms that need to be acknowledged.

You can see and acknowledge active alarms in this list. When these alarms are inactivated, they will appear in the alarm history. Both the reset and the acknowledgment is stored in the alarm history.

7.2 Alarm Types

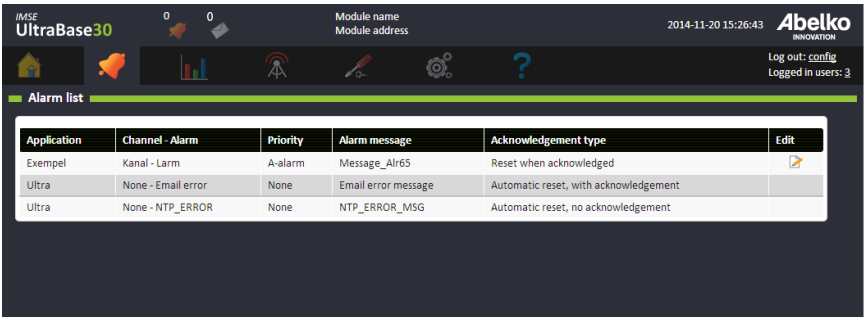
An alarm is created based on a channel in the graphical programming and can have different types of priorities. **A-alarms** have a red indicator and **B-alarms** have a yellow indicator. They can also have a priority from **C-Z**, which have blue indicators. Alarms can have different types of acknowledgments.

In addition to these types, there are **events/errors** and **messages**. **Events/errors** are created by the system and provide information when there is an error in the system, preventing from working properly. **Messages** are created by script and provide information concerning system changes, such as the system being switched over to summer mode.

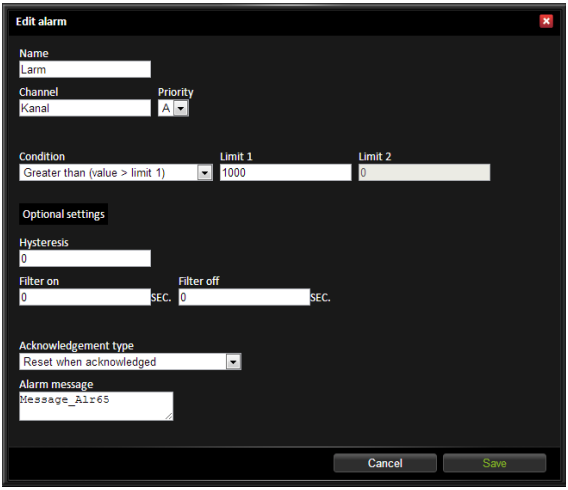
7.3 Edit Alarms

An overview of all the alarms is found and editable under the menu **Alarm**, **Alarm list**.

A complete list of all systems resources is found under the menu **Configuration**, **Applications & resources**.



Click **Edit** to view and change alarm settings.



You can change the **name** of the alarm, see which channel it belongs to and change conditions and alarm limits. Under **Optional settings**, you can edit **Hysteresis** and **Filter**.

The different **conditions** are:

- **Equals (value = limit 1)**: the alarm is activated when the value is exactly the same as the value entered under Limit 1.
- **Greater than (value > limit 1)**: the alarm is activated when the value is greater than the value entered under Limit 1.
- **Less than (value < limit 1)**: the alarm is activated when the value is less than the value entered under Limit 1.
- **Greater (abs (value) > limit 1)**: that the alarm is activated when the value is greater than the value entered under Limit 1, without taking into account whether the value is positive or negative.
- **Smaller (abs (value) < limit 1)**: the alarm is activated when the value is smaller than the value entered under Limit 1, without taking into account whether the value is positive or negative.
- **Between (limit 1 < value < limit 2)**: the alarm is activated when the value is between the values entered under Limit 1 and Limit 2.
- **Outside (limit 1 < value or value > limit 2)**: the alarm is activated when the value is below the value entered under limit 1 or above the value under Limit.

The **Hysteresis** settings prevent the alarm from repeatedly switching between active and inactive when the value is equal to the limit. It specifies the number of degrees by which the value must fall back below the limit or limits before the alarm can be cleared.

The **On-filter** specifies the number of consecutive seconds the alarm condition must be true before the alarm is triggered. The **Off-filter** specifies the number of consecutive seconds the alarm condition must be false before the alarm is cleared.

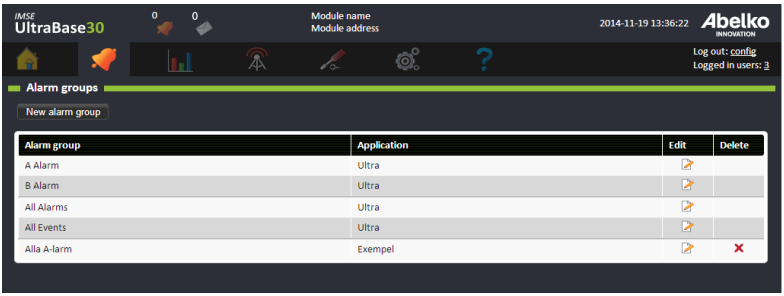
The different **acknowledgments** are:

- **Reset when acknowledged**: the alarm remains active until a user acknowledges it. If the error remains after the alarm is acknowledged, it will remain active until the error is cleared, including the delay and hysteresis.

- **Automatic reset, with acknowledgment**: the alarm reset itself when the error is cleared, but it still has to be acknowledged. In this case, the acknowledgment is a kind of confirmation that someone has noticed the alarm. Details of the reset and the acknowledgment are both recorded in the alarm history.
- **Automatic reset, without acknowledgment**: there is no need for a user to acknowledge the alarm. The alarm is cleared when the error is no longer active as defined in the alarm conditions. You can specify delay and hysteresis values in the condition.

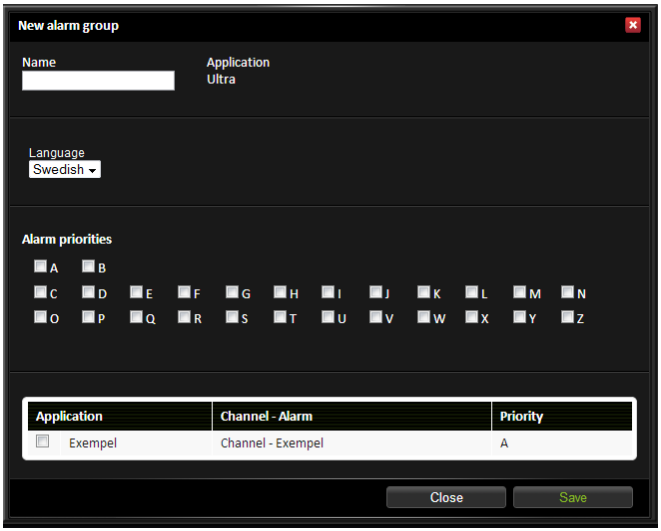
7.4 Create Alarm Groups

Create alarm groups by grouping alarms from an application.



1. Create a new alarm group by clicking **New alarm group**.

2. **Name** the group and choose which language the sendouts will be in.



3. Mark the alarms you want in this group. The benefit of using this method of creating alarm groups is that you can choose alarms from the entire system.

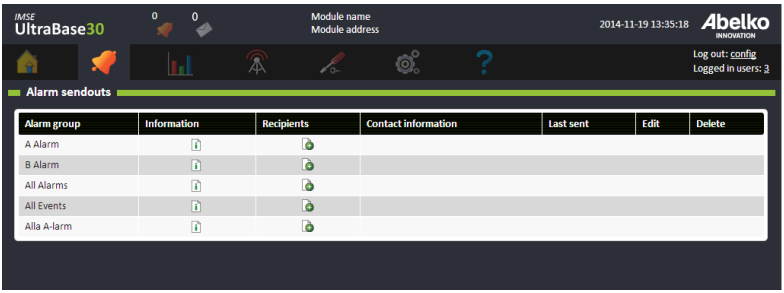
4. Future alarms can be automatically added to created alarm groups. For example if you want all the **A-alarms** in the system to be part of a group, check the box next to the **A**. By doing this, all future **A-alarms** created in the system will also be included in the alarm group.

5. Click **Save**.

To set up mail and text sendouts for alarm groups, go to the **Alarm sendouts** menu.

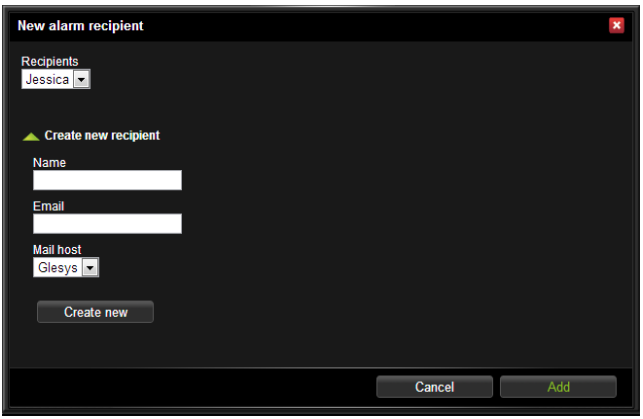
7.5 Alarm Sendouts

Here you can create and edit alarm sendouts.



1. Alarm groups must be created before editing alarm sendouts (see previous chapter).

2. Click add new under **Recipients**. Choose a recipeint in the list (these are created under the menu **Communication**, **Sendouts**).



3. If you want to **add a new recipients** but don't have a **Mail host** you must first fill out the **Mail host** settings under the **Communication** menu.

7.6 Events/Errors and Messages

Email error	An error occurred when an email was about to be sent. This may be due to incorrect settings or communication error. Check email settings to verify **Server** etc. are correct.
NTP error	The system has not received any time from NTP, which means that the system cannot confirm that the time is right. One of the reasons may be that it has the wrong server settings for NTP or there is something wrong with the NTP.
System restart	This event indicates when the unit has been restarted. You can see when the unit was turned off (the event becomes active) and when the unit started again (the event becomes inactive).

8 Data Logs

Logs are history of what has happened in the system, and are used to create statistics. This chapter will explain how to create and edit logs. There are two types of logs: smart logs and periodic logs.

Logs are created under channel settings both in the **graphical programming** and under the menu **Configuration**, **Applications & resources**.

8.1 Smart Log

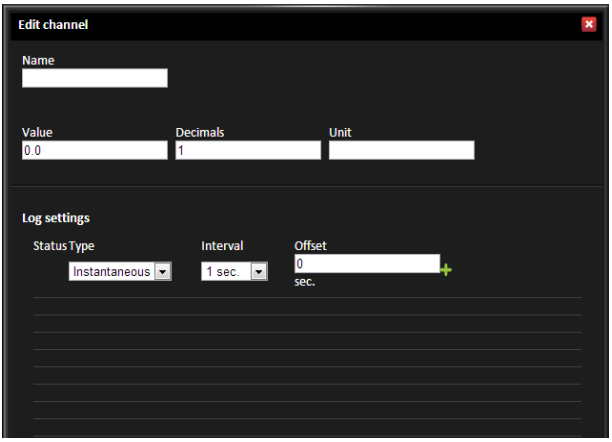
Smart log is a log that compresses the data that comes in. Second values are converted to minute values, and so on. This allows you to store data for a longer period.

8.2 Periodic Log

Periodic logs allow you to measure momentary values, which is an instantaneous value. You can monitor statistics by measuring **mean value**, **max value**, **min value** or a **sum of a certain period**. Then select the **range** you are interested in (1 second, 2 seconds, 15 seconds, 1 minute, 5 minutes, 15 minutes, 1 hour or 12 hours). If you don't want to set a **range**, select **log when the value changes**. In addition, the change log can be used for a parameter, and it will appear in the log if someone changes it.

8.3 Create logs

1. Go to a channels settings (in the **graphical programming** or under the menu **Configuration**, **Applications & resources**).



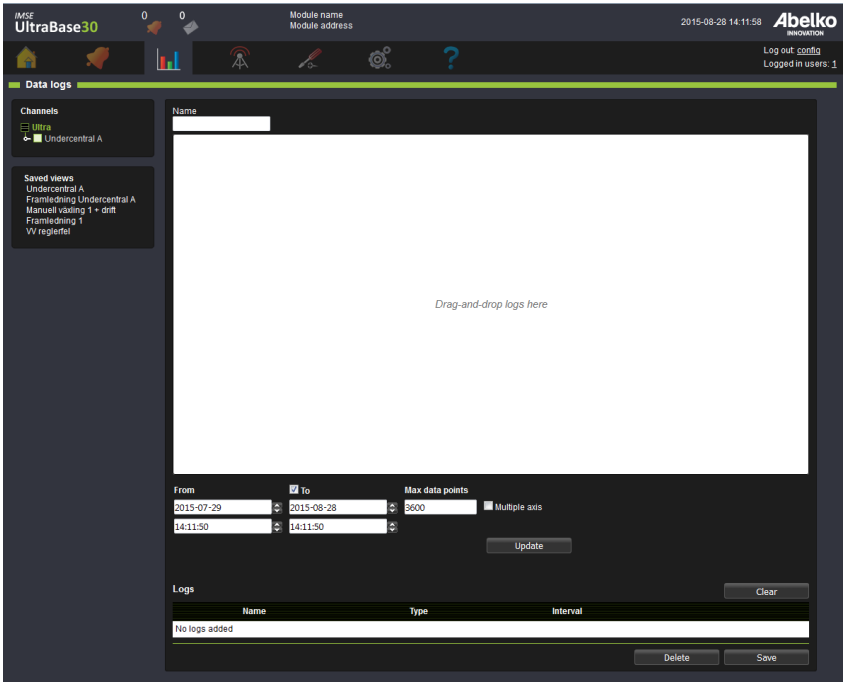
2. Choose **type** of log, set **range** and **offset**.

3. Click on the **plus** to add a log. You can have different types of logs for the same channel.

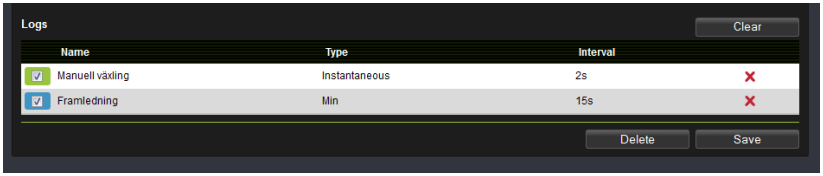
4. Click **Save**.

8.4 View Logs

View the logs under the menu **Data** and **Data logs**.



1. Drag-and-drop channels from the node tree to the left, into the workspace. They will appear in the list under the graph.



2. Change the view settings under the graph.

3. Save the view by naming it, and then click **Save** at the bottom of the page.

9 Inputs & Outputs

In this chapter, you will learn the principle behind inputs and outputs. We will explain how to connect them on the physical unit and also how to configure them in the web interface.

9.1 Overview

For the device to be able to work, you must first connect the different types of sensors and actuators. These are connected to inputs and outputs and are represented in software by I/O channels. An UltraBase30 has its own inputs and outputs, and the number of inputs and outputs can be expanded by attaching expansion modules. Both the expansion modules and the UltraBase30 are represented as I/O units. I/O units and I/O channels are configured under the software menu **Configuration**, **In- & outputs**.

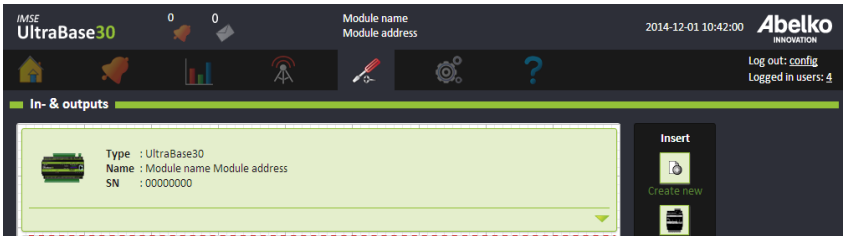
Expansion modules are modules that you plug into an Ultra via ready-made modular cables on a special bus. They are a part of the product line and work the same way as the base units I/Os. They are designed to be mounted in the same cabinet as the base unit. Up to ten units can be connected, and they have cable sequence detection that detects the order in which they are connected. Cable sequence number is used to address the device, but you can also use the serial number as the address.

In- and outputs can also be connected via **external units**, which communicates via Modbus or other protocols. You can read about this in more detail in Chapter 10.

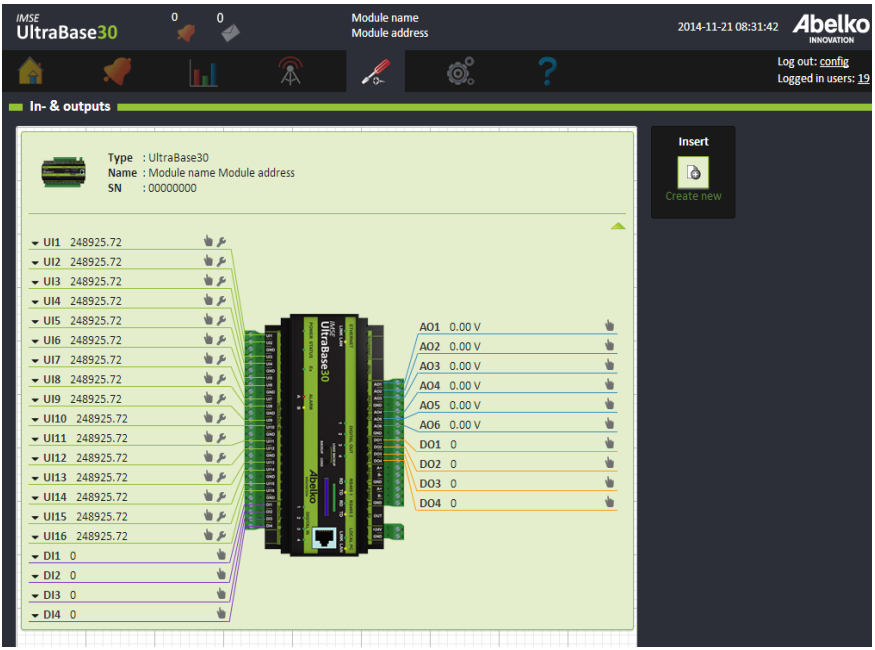
9.2 Configure Sensors and Actuators

Edit all of the units inputs and outputs under the menu **Configuration**, **In- & outputs**, including expansion modules if any.

The control unit is located at the top of the page.



1. If you open an unit online, you can see all of the inputs and outputs.



2. Click on the inputs and outputs **placement names** (ex. UI1) to change their settings (number of decimals, unit, scaling etcetera) .

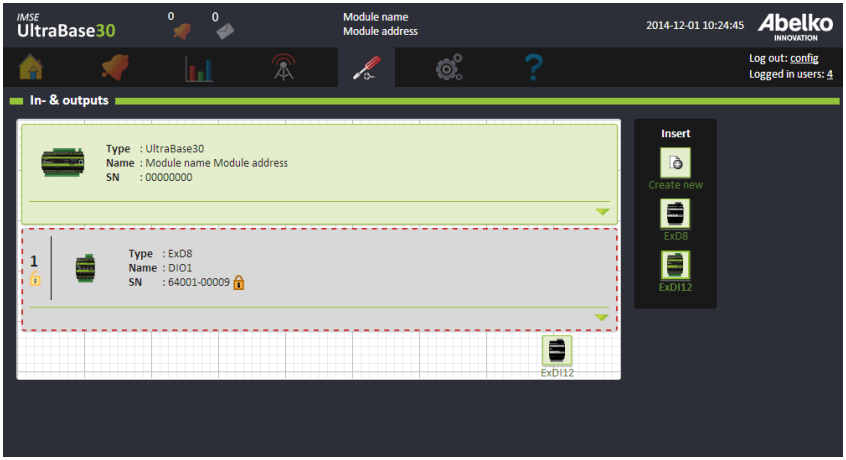
3. A **small arrow** next to the placement name indicates that you can use the same input multiple times, measuring different types of values. Click on the arrow to change their settings.

This icon means that there are settings concerning the input/output itself. For universal inputs, select if you want to measure resistance, voltage, current or digital status.

This icon means that the channels can be controlled manually (manual override).

9.2.1 Configure Expansion Modules

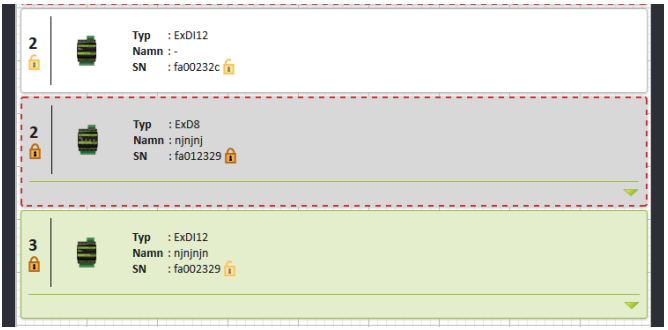
You can configure expansion modules in two ways. First, by connecting the expansion modules and then editing their settings once they appear in the web interface. Or second, by preconfiguring the expansion modules before they are connected to the UltraBase30; if you preconfigure them, prepare the modules in the same order you want them to be connected physically.



1. Drag-and-drop modules from the **Insert** menu into the workspace. When you drop the module, it will appear as a block in the list.

2. The modules are numbered in the order you place them in the list. This order represents the order that they should be connected physically to the base unit. You can easily change the order by dragging-and-dropping the blocks up and down in the list.

3. **Lock** the modules to a **cable sequence number** and/or a **serial number** so they can be used in the system. White background and a grey frame means that you have not yet locked the module. When you preconfigure an Ultra, you lock to the cable sequence number. If the correct type of module is located in the right position, everything will work properly. When the module is connected, lock to the serial number; that way the connection is made to that specific module, regardless of position.



4. When you physically connect the modules to the base unit, their positions must be matched to the order you prepared the modules for. If there are any **mismatches**, the software blocks for the modules will turn grey with a red dotted frame. To correct mismtches, **reorder the modules** by moving them in the list or change the serial number of the blocks in the interface.

5. If you try to connect a module unfamiliar to the system, you will be unable to make any settings for it and only view it in the list. In order to be able to use it, you must upload a communication script for this unfamiliar module.

Universal Inputs

There are 16 universal inputs. These can function as:

- Analog voltage input: 0-10 V
- Analog current input: 0-24 mA
- Resistance input: 80-200k Ω (for temperature measurement)
- Digital status: for potential-free sensors

Select which function you want them to have under the wrench icon.

Digital Inputs

There are 4 digital inputs. These can function as:

- Digital status
- Frequency measurement
- Pulse counter

This information is available on different channels and can be used simultaneously.

Analog Outputs

There are 6 analog outputs.

- Output signal: 0-10 V
- Max output current: 2 mA

Digital Outputs

There are 4 digital outputs.

- Output: Open drain
- Max current: 0,4 A
- Max voltage: 36 V

9.3 Replacing an Expansion Module

You can only replace modules that are not being used. Once you started using a module in script, it is complicated to change it since it can interfere with the script.

1. Go to the menu **Configuration**, **In- & outputs**.

2. Click on the **lock** icon on the module you want to replace. If it is being used, you need to go to the graphical programming and delete all its connections. If it is not being used, you can unlock it and replace it.

3. Connect the new module. The system will detect it, and you can place it in the list over the modules. **Lock** the module to a position and/or a serial number.

4. The new module is now ready to be used in script.

10 External Units

External units are different units that an Ultra can exchange information with as the master unit. They can be used for more inputs/outputs, indicators and more. This chapter will explain how to add new devices and how to access an overview of available units in the system.

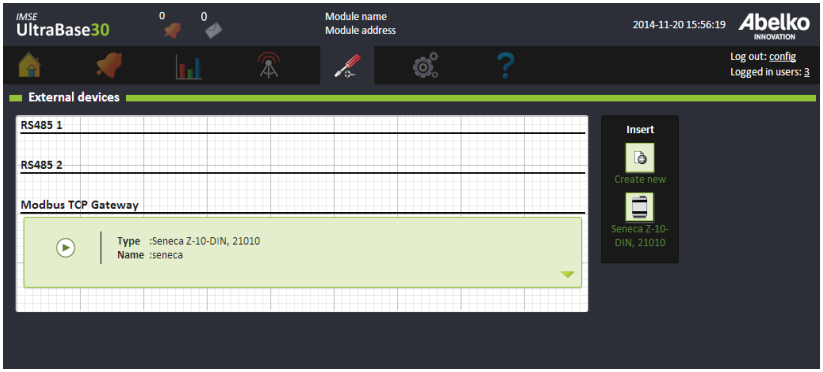
External units can be configured in the graphical programming before they are connected to the Ultra.

10.1 Configure External Units

To the right, there is an **insert** menu with all external unit templates that you have script for.

1. Click **Create new** under the **Insert** menu to upload a new type definition. Select a type definition from files that have been uploaded to the folder **Communication definitions** on the page **File manager**. You can upload type definitions from abelko.se. Type definitions written for IMSE WebMaster Pro are also compatible.

2. Name the new type.

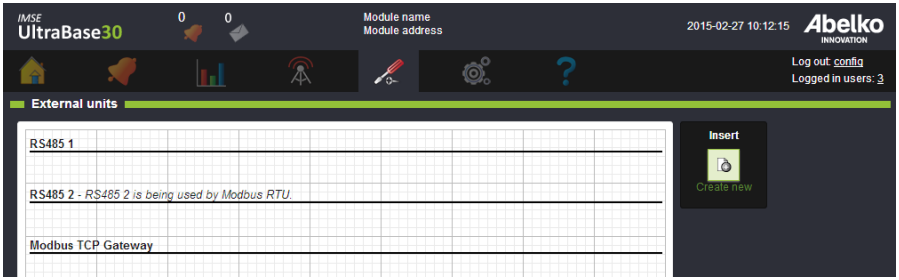


3. Drag-and-drop the template onto the workspace. Place it under **RS485 1**, **RS485 2** or **Modbus TCP Gateway**.

4. Hover your mouse above the block to **Edit** update times. You open the block by clicking on the arrow to make settings and view information about the block.

5. The block will be green when communication between the master and the external unit is OK. However, if something is wrong, the block will turn grey with a red dotted frame. Check that the external device is correctly connected and has the appropriate settings.

6. The external unit is ready for use.



RS485 2 can be used by **Modbus RTU** (this is a setting under the menu **Communication**, **Modbus slave settings**). External units placed under RS485 2 cannot be used when Modbus RTU is activated.

External units located under **Modbus TCP Gateway** send their telegrams to an internal Modbus TCP gateway. You can create and edit rules for these under the menu **Communication**, **Modbus TCP Gateway**.

10.2 Status

The external units can have different statuses that are indicated by the color of the blocks.

- When everything is OK, the block is green.
- If there is something wrong with communication between the master and the external unit, the block turns grey with a red dotted frame.

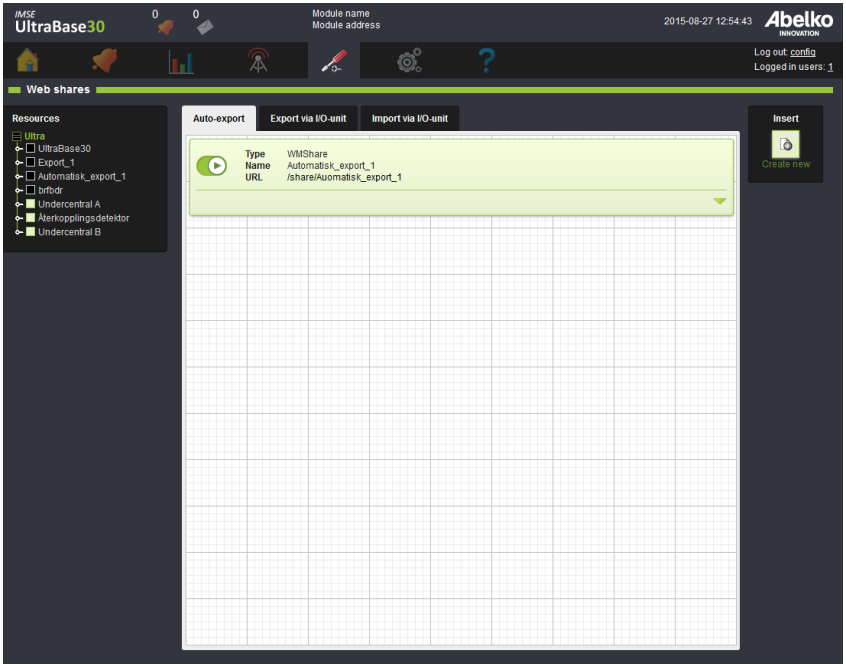
11 Web shares

Web shares makes it possible to communicate with other devices and systems via network-based protocols. In the basic version only WMShare is supported, which primarily is created to share values between Ultra or WMPro. WMShare can also be used against servers and other systems.

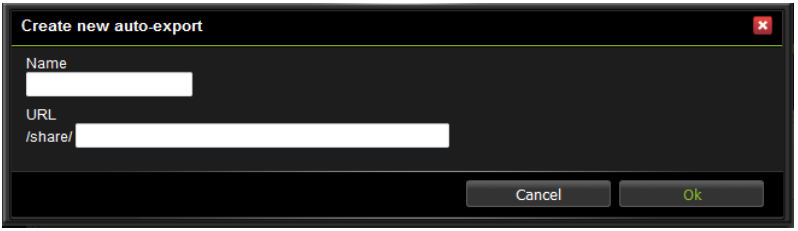
With WMShare export you can distribute the values of other devices, which retrieves them over the Internet like a normal web page. With the import function, you can retrieve values to your Ultra. Each value has a key which is the name of the value.

11.1 Create an Auto-export

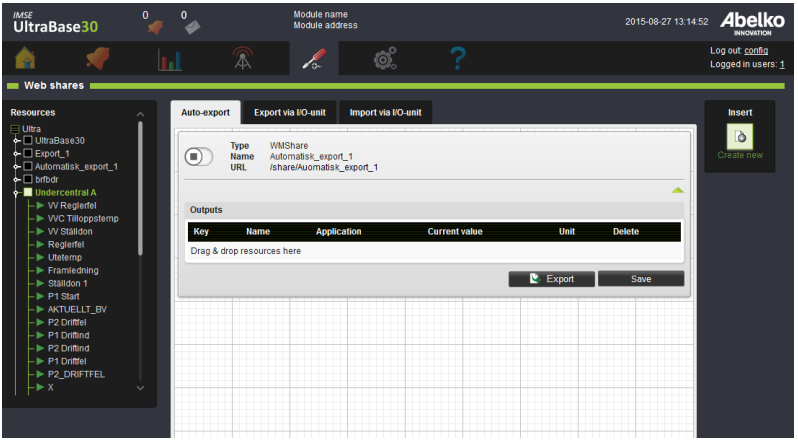
An auto export means that you share values to other units directly without connecting values to an external unit and connections in the graphical programming.



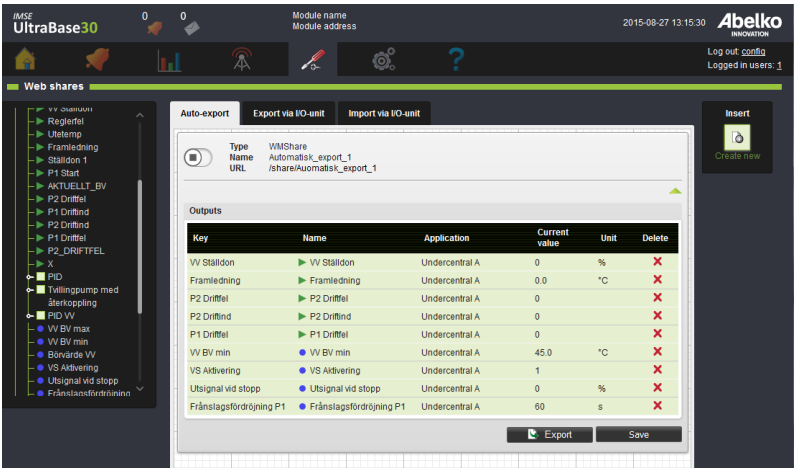
1. **Drag-and-drop Create new** from the **insert** menu to the right onto the workspace to create a new auto-export.



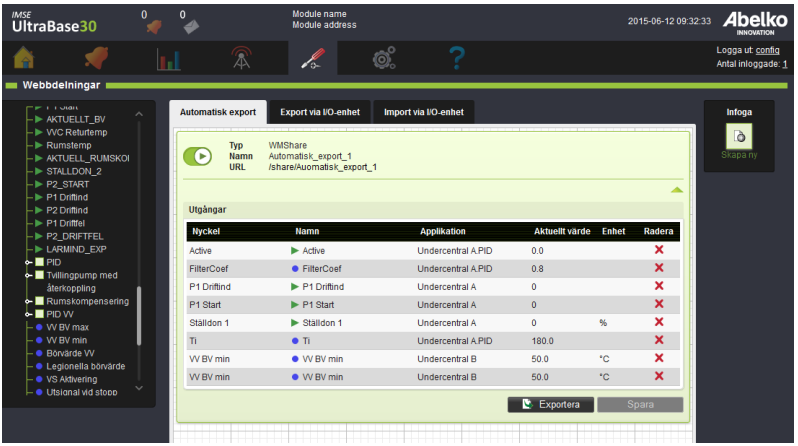
2. **Name** the export. Use underline instead of space since the URL-address will get the same name. Click **Ok**.



3. **Drag-and-drop** channels, parameters and more from the node tree to the left onto the area under **Outputs** to add them.



4. Click **Save**.



5. Click on the status symbol in the upper left corner of the block to **activate** the share. The block turnes green with a play symbol.

Now others can use these values by importing the share with the help of the URL address.

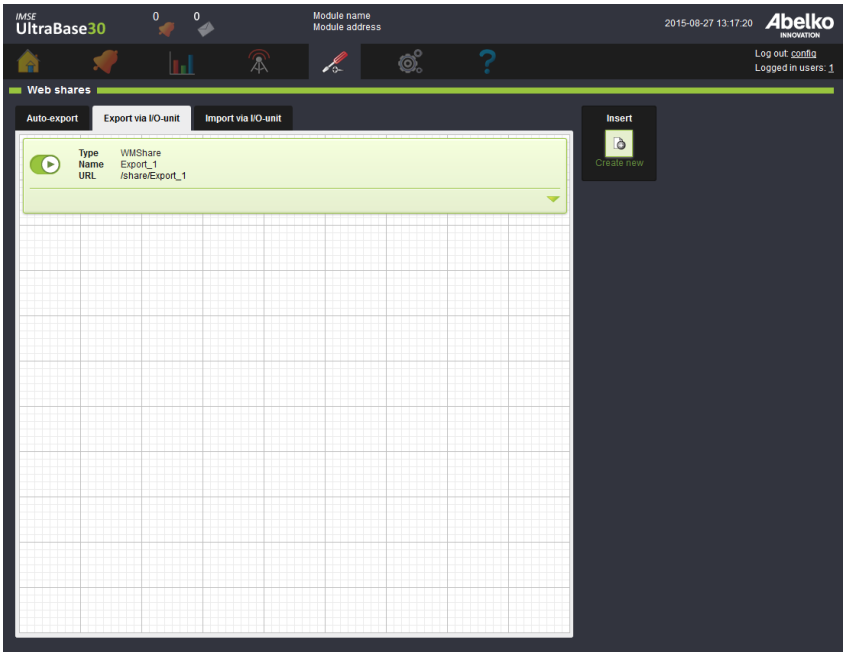
If you click on the URL-address an WMShare-file opens in a new tab.

As default, the key name is the same as the channel name, but you can click

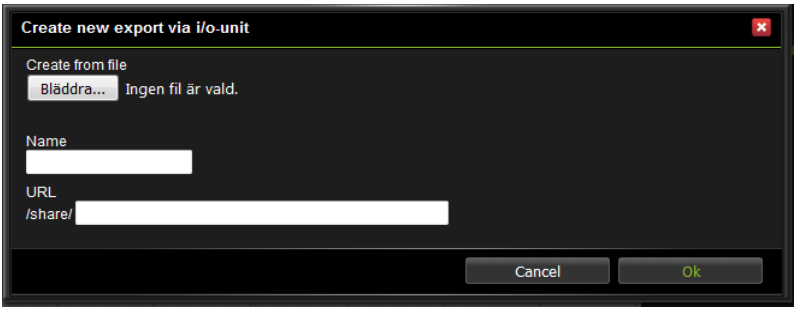
and change the name of the key.

11.2 Create an Export via I/O-unit

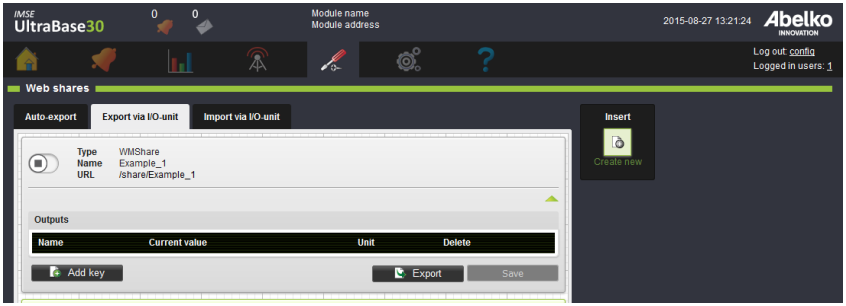
Export via an I/O-unit makes it possible to share values, just as an automatic export. The difference is that it will look like an I/O-unit with outputs that you connect in the graphic programming.



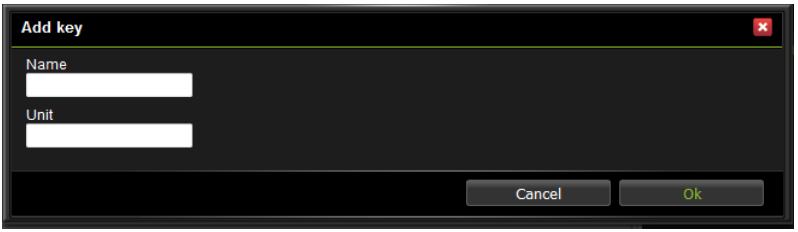
1. **Drag-and-drop Create new** from the **insert** menu to the right onto the workspace to create a new export.



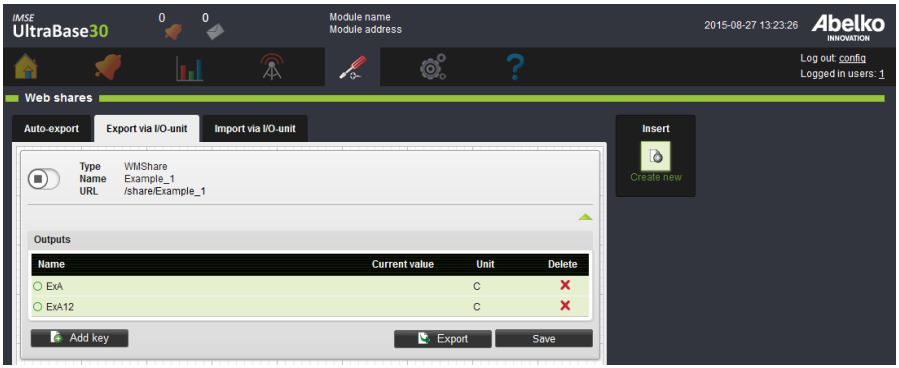
2. Enter a **name** and an **URL** address.



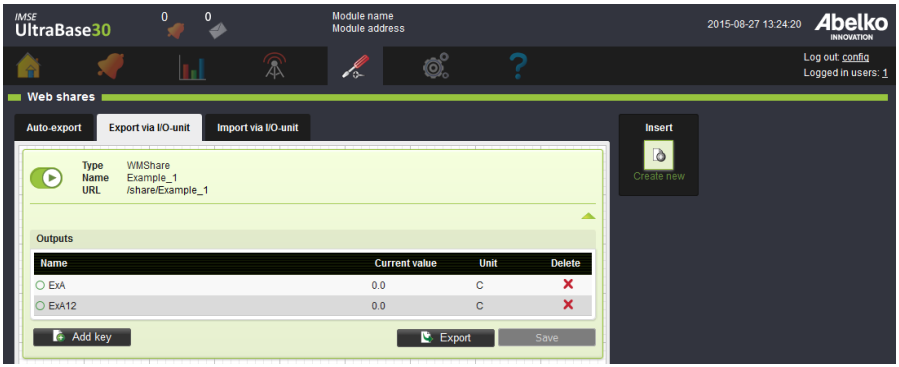
When you create an I/O-unit, you can use a template file. The template file is obtained from the export button in a WMShare. This will fill in the key names automatically. If you don't have a template file, you can manually fill in the key names later.



3. Add, delete or edit key names. When you click save, the key names will turn into I/O-channels. The name should not contain å, ä or ö.



4. When you added all the keys, click **Save**.

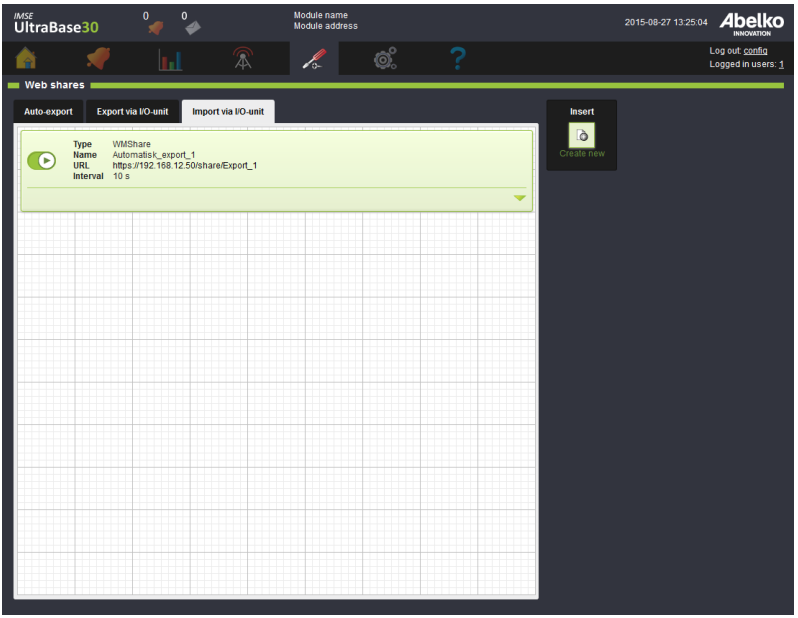


5. Click on the status symbol in the upper left corner of the block to **activate** the share. The block turnes green with a play symbol.

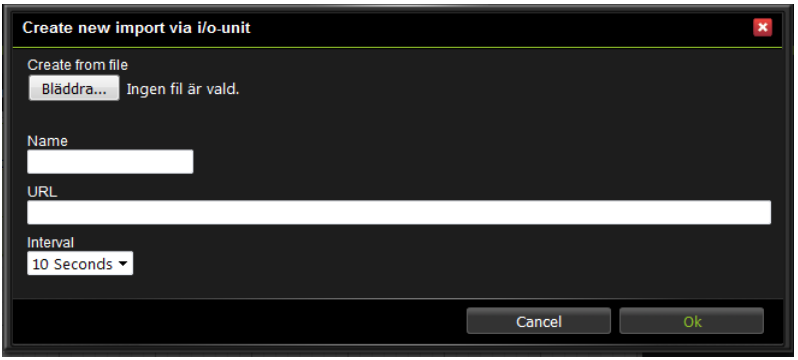
Now you can go to graphical programming and connect the external units outputs.

11.3 Create an Import via I/O-unit

In order to import values from another unit, you create an import via an I/O-unit.



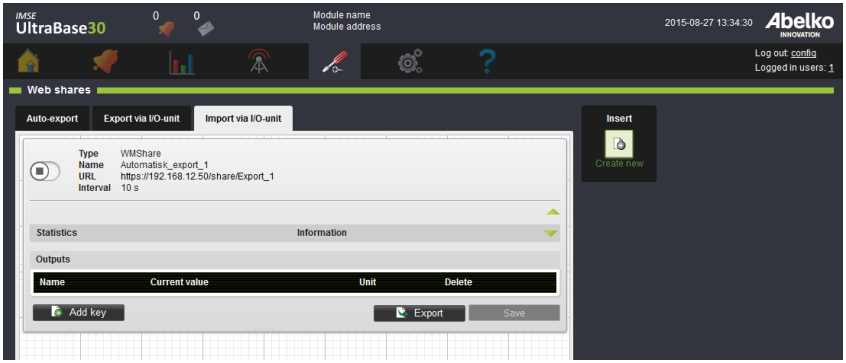
1. **Drag-and-drop Create new** from the **insert** menu to the right onto the workspace to create a new export.



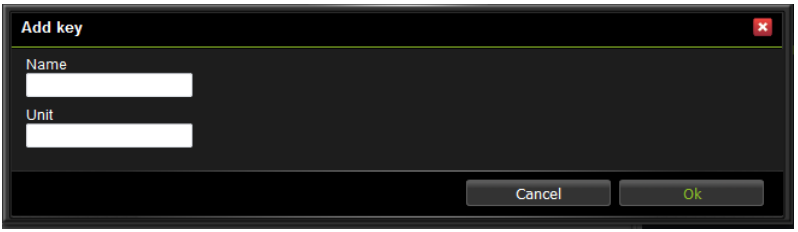
2. When you create an import, **name** the I/O-unit and enter the **URL** address to the share you want to import. Under **Intrevall**, you set how often you want the values to be updated. Click **Ok**.

If the sharing unit already is running and operating, the key names can be collected from there. Else, you can use a template file. The template file is

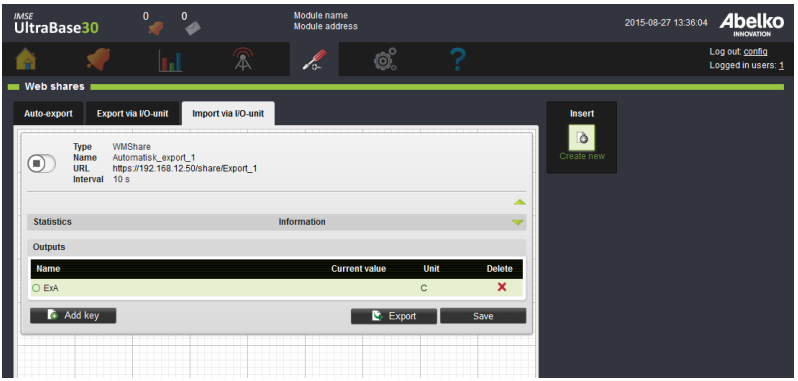
obtained from the export button in a WMShare. This will fill in the key names automatically. If you don't have a template file, you can manually fill in the key names later.



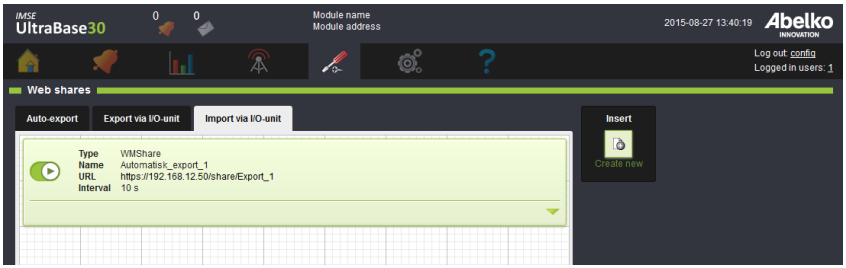
3. Add, delete or edit key names. When you click **save**, the key names will turn into I/O-channels. The name should not contain å, ä or ö.



4. When you added all the keys, click **Save**.



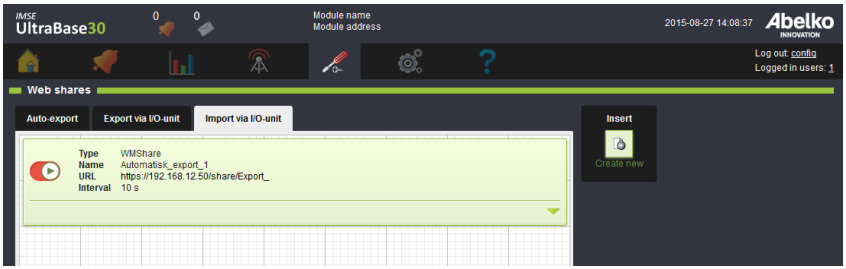
5. Click on the status symbol in the upper left corner of the block to **activate** the share. The block turnes green with a play symbol.



The share is now presented as an external unit in graphical programming where you can connect the units inputs.

11.4 Error Status

If a blocks frame turns red it means that the communication does not work properly. Either the share cannot be retrieved from the specified URL, or there are some keys missing in the retrieved share.



If any key is missing, the row turns red.

For an import, you can see under statistics the amount of retrieval attempts, amount of succeeded attempts and when the last successful import occurred. If a key is missing during an import, it will count as a format error.

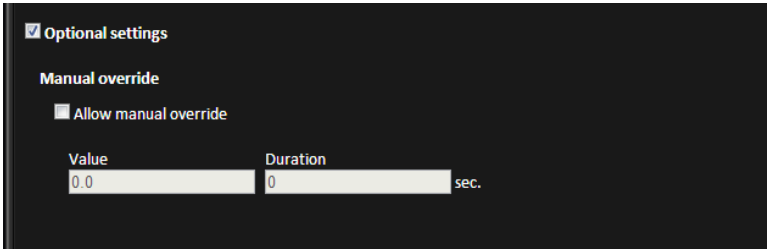
12 Manual Override

Manual override means that you manually can force a channel to obtain a certain value. Manual override can be used on inputs, outputs, and internal channels. This is useful to test applications or a plant before putting it into service.

12.1 Allow Manual Override

In order to use manual override, you must first activate the function for the channel.

1. Go to a channels settings (in the **graphical programming** or under the menu **Configuration Applications & resources**).



2. Check the box **Allow manual override**.

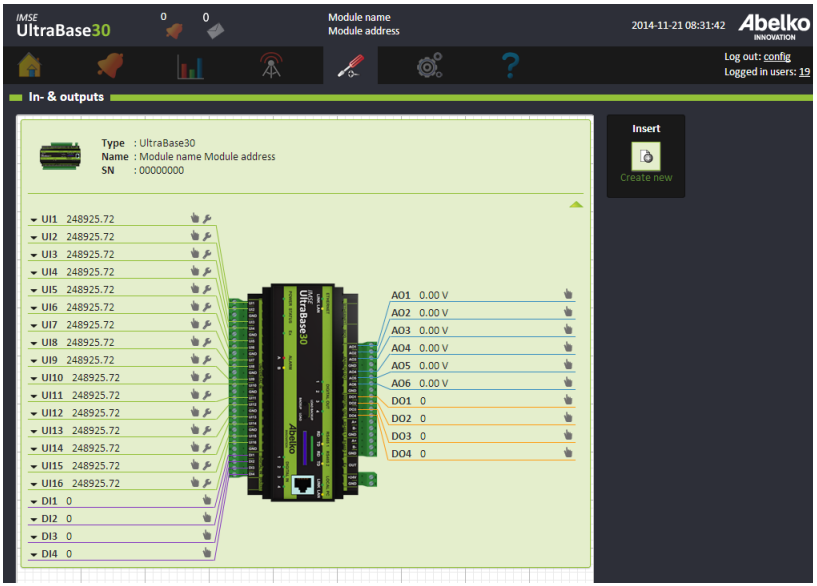
3. Enter **Value** and **Duration** (for how many seconds the channel will have this value) you want the channel to have.

4. Click **Save**.

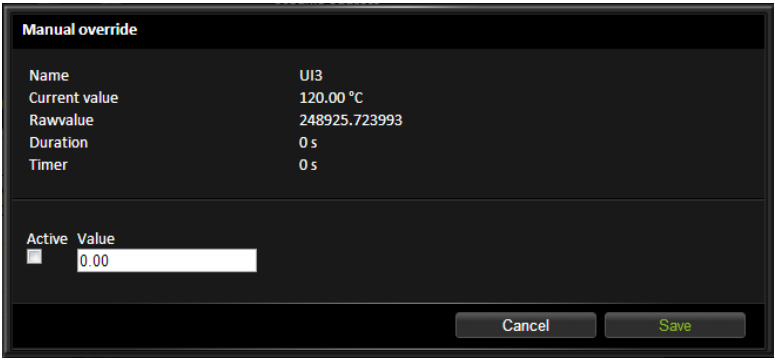
12.2 Activate Manual Override

When manual override is allowed for a channel (see previous chapter), go to the menu **Configuration**, **In- & outputs** in order to activate it.

1. Click on a unit in order to see more detailed information.



2. Channels who allows manual override will have a hand symbol next to it. Click on the **hand** symbol.



3. Here you can edit the manual value. Check the box **Active**.

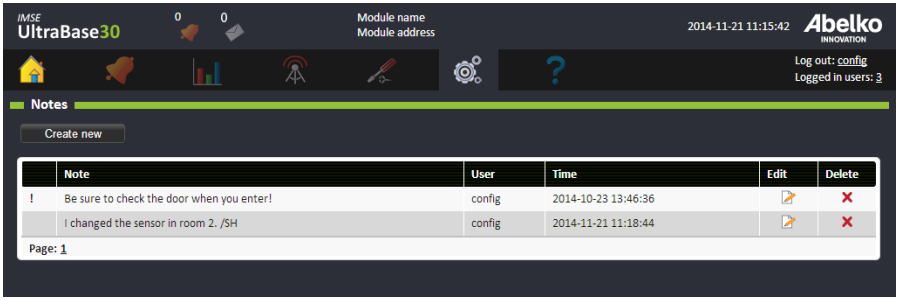
4. The channel will get the manual value when you click **Save**. The duration counter will start the countdown and when it reaches zero, the channel value will return to its real value.

13 System

This chapter will explain the note tool and the file manager. You will also learn how to create backups and update the system.

13.1 Notes

In **Notes**, you can create notes and see changes made in the system. If you have made changes in the system that you want the other users to know of, you can create a note about it here. A new note is indicated by the mail icon on top of the interface for the other users. Click on the icon ito view the note or open the menus **System** and **Notes**.



1. Click **Create new**.

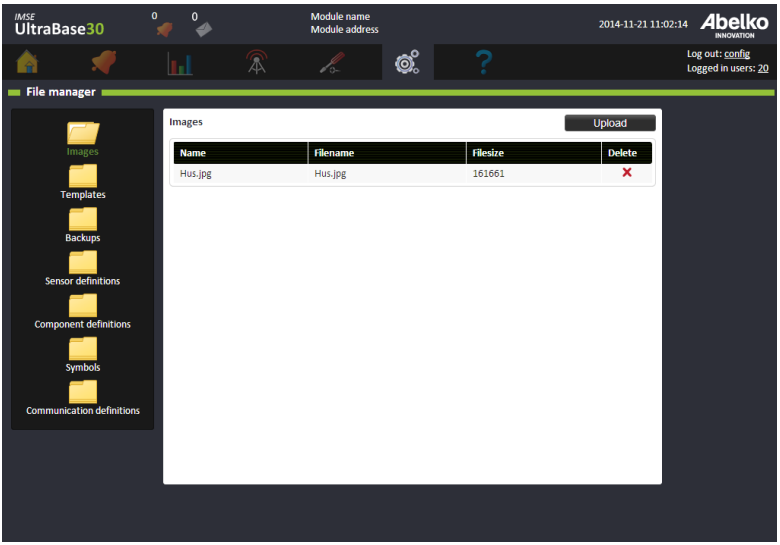
2. Add a note and your name. You can mark the note as **High priority**. This means that it will be placed at the top of the list with an exclamation mark.

3. When you click **save**, all users will have a new note to read.

Click on a note in order to read the whole note.

13.2 File Manager

All system files, including SD card files, are available in the **File manager**. You can copy a file to the PC or the SD card from the system.



Upload files to the system through the predefined folders.

Images: upload images for overviews and other pages. The recommended formats are jpeg, png and gif.

Templates: upload application templates that were created in an Ultra.

Backups: upload application backups containing parameter banks (no user settings, IP address or network settings) from an Ultra.

Sensor definitions: upload definitions for sensors and actuators.

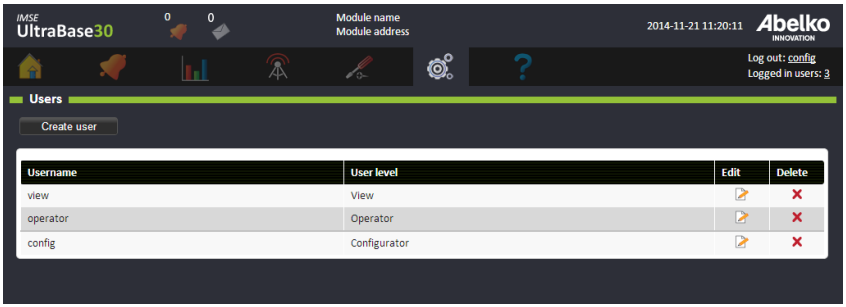
Component definitions: upload components to use in the graphical programming.

Symbols: upload symbols to use when you create overviews.

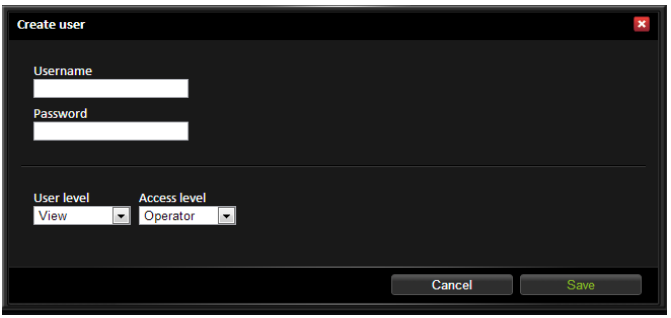
Communication definitions: upload definitions for external units and expansion modules. You can use various communication definitions and Abelkos own software, M-bus device creator. Usable definitions are available at script.abelko.se.

13.3 Users

Create and edit users.



1. To create a new user, click **Create user**.



2. Add a **username** and a **password**. Select a **user level** and also an **access level** (the user level required to edit the user).

3. Click **Save**.

User levels

Configurator: has access to all the settings of the system.

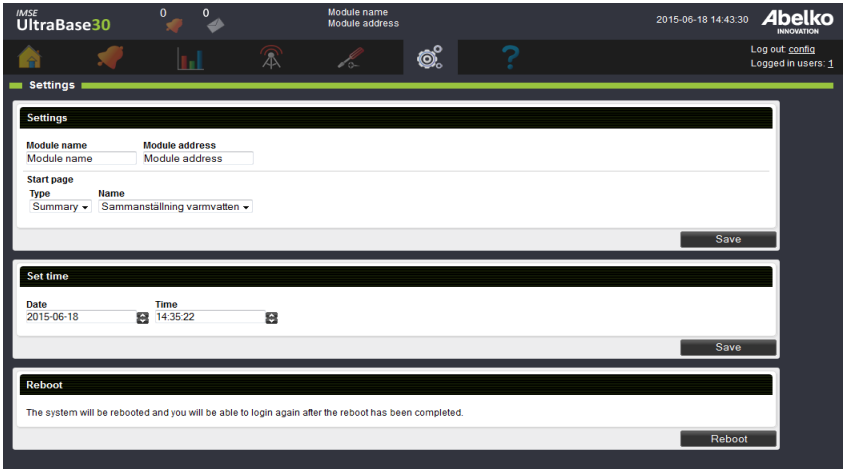
Operator: adjusts set points, acknowledge alarms and more, but does not create new applications or upgrade the device.

View: cannot change settings.

13.4 Settings

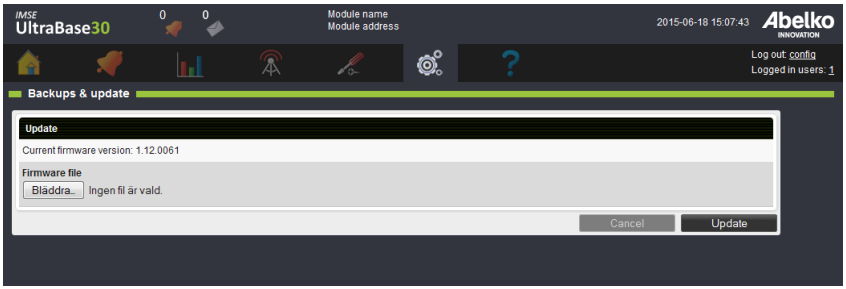
Enter the name and the address of the module under the menus **System** and **Settings**.

Here you select a **start page.** Choose between overviews and start pages. You can also **reboot** the system if needed.



13.5 Backups and Updates

To update using the interface, go to the **Backups & update** menu under the **System** menu.



1. Select a firmware file.

2. Click **Update**. **Note** that you can also create a backup through the

physical unit.

3. You will soon see a new window with update information.

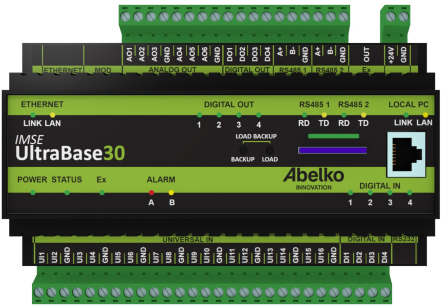
4. When the system has updated everything, you will be able to **log in** again.

Note! SD-cards are not recommended for long term data storage. It is safer to transfer the data to a computer or similar.

13.5.1 Via the Unit UltraBase30

At the front of an UltraBase30, there is space for a SD card.

Note! If someone else is working on the device through the interface, the light above the SD card will flash green. Do not touch the SD card or the backup buttons during this process since it may interrupt their work.



Create a full backup

1. To create a full backup of the systems data (settings including network settings and databases with logs - a copy of the unit) to the SD card, hold the **BACKUP** button for **3 seconds**.

2. The light above the SD-card will flash green during the process.

3. When the process is complete, the light will shine green for **5 seconds**. Now the backup is finished.

If an error should occur, the light will flash red for 15 minutes. A file will be transferred to the SD card with information concerning the error (as long as the card is not write-protected).

Note: SD-cards are not recommended for long term data storage. It is safer to transfer the data to a computer or similar.

File transfer

To transfer files, the file should be a zip file and can contain sequence files, clone backup (a backup without databases), software upgrades, parameter bank scripts etc.

1. To transfer a file, hold the **LOAD** button for **3 seconds**.

2. When the process is complete, the light will shine green for **5 seconds**. Now the file is transferred.

If an error should occur, the light will flash red for **15 minutes**. A file will be transferred to the SD card with information about the error (as long as the card is not write-protected).

Load a full backup

You can load a full backup into the system (settings including network settings and databases with logs). The backup is usable when you, for example, want to replace an unit and need an exact copy.

1. To load a full backup, hold the buttons **LOAD + BACKUP** for **3 seconds**.

2. When the process is completed, the light will shine green for **5 seconds**. Now the full backup is done.

If an error should occur, the light will flash red for 15 minutes. A file will be transferred to the SD card with information about the error (as long as the card is not write-protected).

13.6 Information

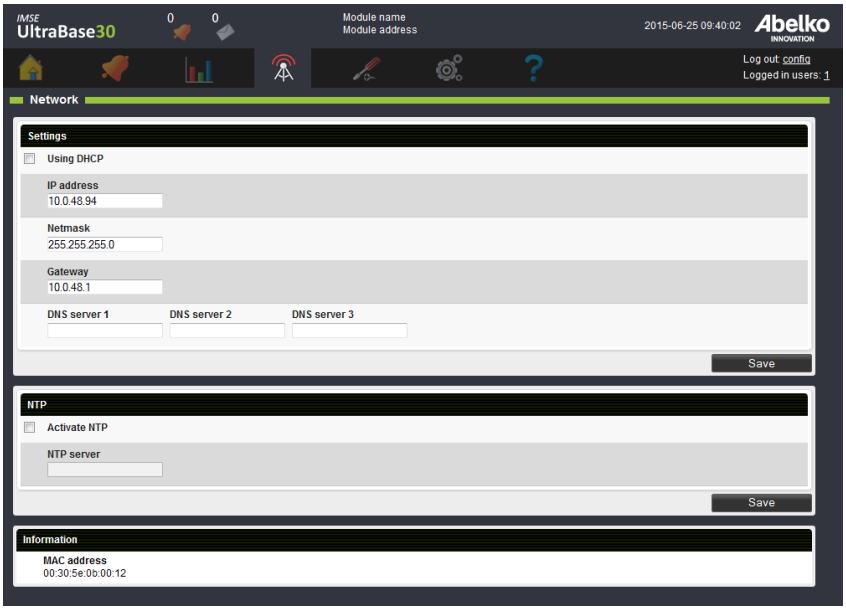
Under the menu **System**, **Information**, you can view information about the system and its operation.

14 Communication

Under the **Communication** menu, you can view an overview of what type of information the different recipients will receive, create register lists and edit mail, texts, network and Modbus.

14.1 Network

Network allows you to handle network settings.

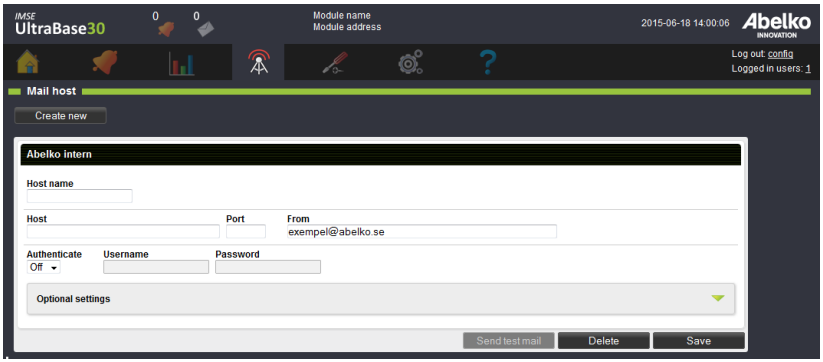


If you change the IP address and click **Save**, you will be logged out from the system. Enter your new IP address up in the address bar in order to log in again.

14.2 Mail Host

Here you can edit the settings for the unit's mail host. This setting is required in order to send emails for both alarm and log sendouts.

1. Enter **Host**, **Port** and a valid **email** for the mail host.



2. **Host name** is used only if you want to be able to distinguish the different servers in the system.

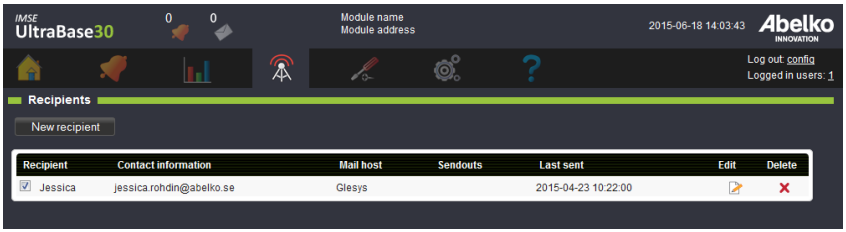
3. Many email servers require **authentication**. Select **Auto** and enter **Username** and **Password** for the email server (this is something you receive from your Internet service provider).

4. Click **Save**.

All email recipients are tied to an email server defined in the settings. You can define more than one email server.

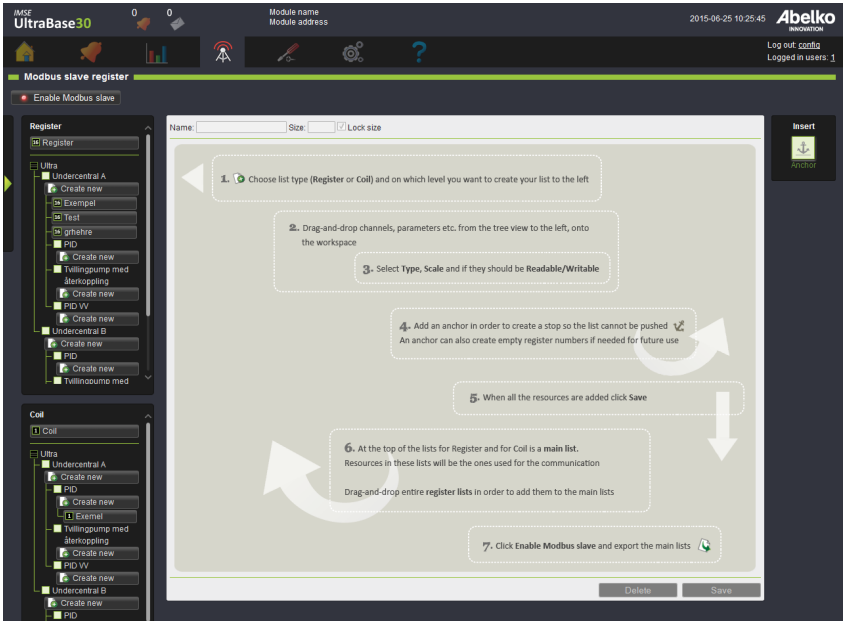
14.3 Recipients

View and edit all the people registered in the system. You can also add new recipients. This section gives you an overview over which alarm and log sendouts the different recipients will receive.



14.4 Modbus Slave Register

Modbus is a protocol used to communicate between devices. Here you create and edit register lists for Modbus-slave-communication.


UltraBase30
Module name
Module address
2015-06-25 10:25:45
Abelko
Log out: config
Logged in users: 1
Modbus slave register
Enable Modbus slave
Register
Coil
Name:
Size:
Lock size
Insert
Anchor
1. Choose list type (Register or Coil) and on which level you want to create your list to the left
2. Drag-and-drop channels, parameters etc. from the tree view to the left, onto the workspace
3. Select Type, Scale and if they should be Readable/Writable
4. Add an anchor in order to create a stop so the list cannot be pushed
An anchor can also create empty register numbers if needed for future use
5. When all the resources are added click Save
6. At the top of the lists for Register and for Coil is a main list.
Resources in these lists will be the ones used for the communication
Drag-and-drop entire register lists in order to add them to the main lists
7. Click Enable Modbus slave and export the main lists
Delete
Save

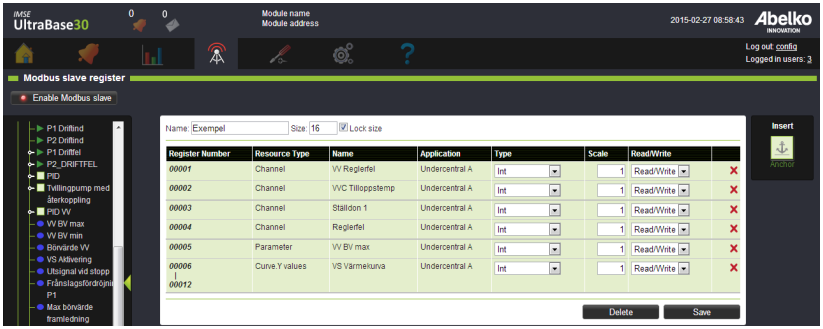
Create a register list

1. Decide which type of list you want to create, **Register** or **Coil**. A register is a 16-bits word and a coil is a 1 bit word.

2. Click **Create new** under the application you want to create from. If you want to be able to use all of the systems resources, create a new list directly under the Ultra.

3. Open the node tree and drag-and-drop the resources you want to have in the list. You can only choose resources from the application you created the list in, and also its subapplications. You can also link another register list in your list by dragging and dropping it from the node tree.

4. The resource will receive a **Register number**. Move the resources in the list in order to change its register number.



5. Select **Type** (16-bits, 32-bits, 64-bits and so on). Since every row is 16 bits, a 32 bit will reach over two rows. The value is multiplied by a scaling factor before it is added to the register. For integer registers you can use this to get decimals. The Modbus master then needs to divide with the same scale factor on its side. For example, with the **Scale** 10 for a temperature, the master will read the value 257 if the temperature is 25.7 degrees. A coil are always 1 bit so they don't have these settings. Also, select if you want it to be **readable**, **readable/writable** or **none**.

6. Click **Save**.

The list located directly under the Ultra is the list with the final addresses. This means that this is the only list that can be exported and used.

7. Click **Enable modbus slave** in order to lock the lists and start the communication.

8. Click in order to export the list. You can export two lists, one for **Register** and one for **Coils**.

To the right there is an **Insert** menu.

Anchor works as a stop so the list cannot be pushed over a set register number. It can be useful to add an anchor in order to save space for future resources so you don't have to change the order of the register numbers.

Change an active register list

When you have a register list in use, changes like adding a new row will have concequences for the communication.

1. Inactivate the communication by clicking the button **Disable Modbus slave for editing**.

2. Edit the resources.

3. If you want to add new resources in the list, you can use the empty rows you added with help of the anchor tool.

4. If you add a new row so the register numbers shift, the row will turn red. If you want it to be available for export you need to change the list seize next to the list name. This will create a shift in the list and the register numbers will be updated. Changes like this needs to be adjusted in the supervising system as well.

5. When you have edit the list, click **Save** and then **Enable Modbus slave** in order to continue the communication again.

14.4.1 Register types

Int and Word

In Modbus, a register is a 16 bits word that can have a value between 0 and 65535. In the interface, this register type is called **Word**. If you want to transfer values with decimals over Modbus, you can use a scale factor. In order to transfer a temperature value with two decimals, set the scale factor to 100. The value, for example 12,34°C, will then be read from the register as 1234. The master who reads it must know how much the value is being scaled.

If the value can be negative it needs to have the register type **Int**. The first bit in the 16 bits word will then be treated as a sign bit. Numbers from -32768 to 32767 can be represented.

If the value is outside of this register types range the master will read it as an incorrect value.

Int32 and Word32

If you want to be able to represent larger values, or values with higher dissolution, you can use two Modbus registers in order to represent a 32 bits word. Set **Int32** or **Word32** as register type. **Int32** can represent numbers from -2147483648 to 2147483647. **Word32** can represent numbers from 0 to 4294967295.

Float

Float is a 32 bit floating-point number following the IEEE 754 standard. **Float** can handle numbers with about 6 significant digits, (almost) regardless of how large or small values it is.

One problem with floating-point numbers on Modbus is that it does not have a standardized order in which the four bytes that make up the floating-point number should be sent. The Ultra supports all of these four possible ways.

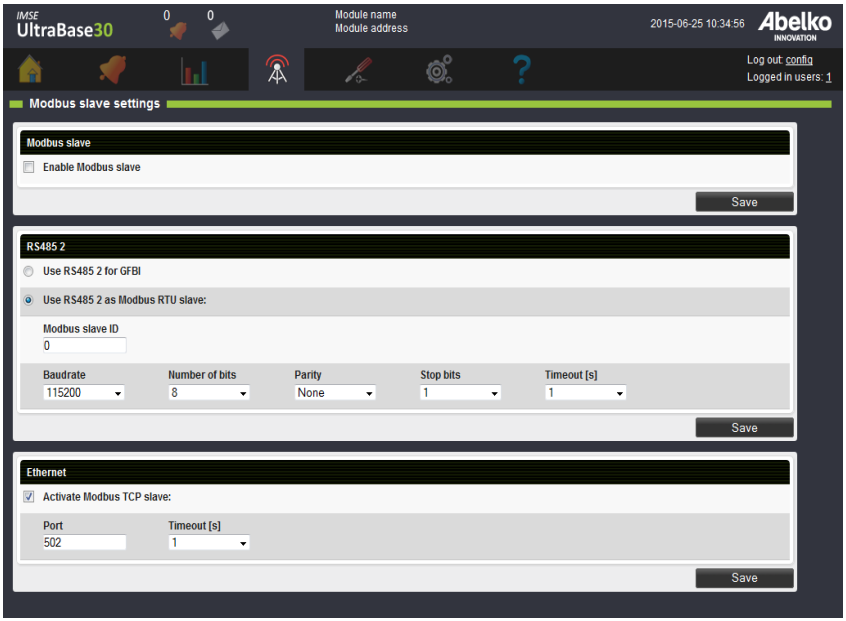
Below you can see how Pi is represented as a float, and how the four bytes titled A to D.

s	exponent								Mantissa																			
0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	1	1	1	1	1	0	1	1	0	1	1
4							0		4				9			0				F				D				B		
Byte D									Byte C							Byte B								Byte A						

The letters after **Float** when you set register type, indicates the order of the bytes in the two registers that is being used. Master and slave have to code the floating-point number in the same way in order to understand each other.

14.5 Modbus slave settings

Here you activate and edit the Modbus slave function.



Modbus slave

An Ultra can function both as a **Modbus RTU slave** on the **RS485 2 port** and as a **Modbus TCP slave**. Define which values you want to be readable for which registers under the menu **Congifuration**, **Modbus slave register** (see previous chapter for more details). Activate the Modbus slave function with the check box. You cannot edit register settings while the Modbus slave function is activated.

RS485 2

The Ultra unit has a communiaction port marked **RS485 2**. This can either be used for external units, which are defined under the menu **Configuration**,

External units, or used for the **Modbus RTU slave** function.

Activate the **Modbus RTU slave** with the check box and set the communication settings for the port. External units on the RS485 2 port will then stop working.

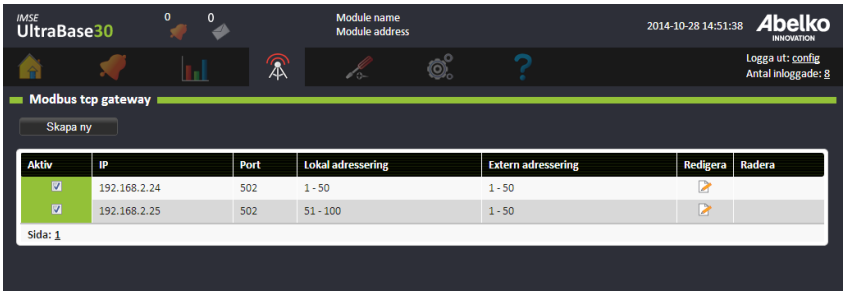
Ethernet

Modbus TCP communication is done over the network port marked **Ethernet**. **Port 502** is the default port number for a Modbus TCP server, but you can select a different port number.

The **Timeout** setting determines how long after the question has arrived the Ultra are allowed to send an answer. The Ultra will respond as soon as it can, but can sometimes be busy with other things. If it has not had time to respond before the timeout expire, it will send an exception kode or does not respond at all. The system's master must wait until the slave timeout expire before sending a new question or send a question to any other device.

14.6 Modbus TCP Gateway

Create rules for the external units located under **Modbus TCP Gateway** on the page **External units**. These send telegrams to an internal Modbus TCP gateway. The rules determine to which IP address and port they should send their telegrams to.



Create new

1. Click **Create new** and enter **IP address** and **port**.

2. Set **Start address** and **Stop address**. This decides which Modbus addresses the rule applies to. The **offset** can be used if you want the address to be translated to another Modbus address for the recipient. This function is usable if the recipient is a gateway to RTU.

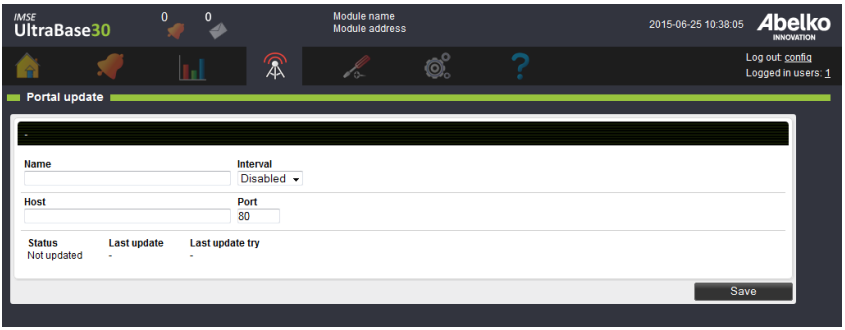
In the example above, we created two rules. The units addressed 1-50 are sent to the address 192.168.2.24. The units with address between 51-100 are sent to the address 192.168.2.25, and the offset (-50) will make the addresses correspond to 1-50 on the recipients RTU bus.

If you only have a Modbus TCP unit to communicate with, you can set the address range to 1-253 and the offset to 0.

3. The rule must be **active** in order for it to work. To activate the rule, check the box on the front page. The Ultra will try to establish a TCP connection to the server as soon as the rule is activated.

14.7 Portal Update

This is a function that, together with a portal server, such as **portal.abelko.se**, helps keep track of the unit's IP address.



Enter a name, and set how often you want the unit to update its address with the server. If you use a mobile subscription or have a connection that changes the IP address frequently, set a short interval. If you have a connection that does not change the address very often, such as fiber or DSL, set a high interval.

On our portal **portal.abelko.se**, you can use the units MAC address, view its current IP address and view when the unit was last connected to the server and updated it. When you found your unit (using the units MAC address), save the result page as a bookmark. This gives you a link that is updated with the most recent address as long as the unit is connected to the Internet.

Please note that the portal service only gives you a link to the product, you do not surf through the portal.

14.7.1 Status

Under status, view when the last update and last update attempt was made. Normally, these two dates are the same, but when the unit fails to update, these dates differ and the status will be **Failed**. Statuses can have the following values:

- **Inactive**: the function is not activated.
- **Updated**: the function is activated and the last update succeessful.
- **Update error**: the function is activated but the last update failed. The most common reason why the device fails to update is that it does not connect to the Internet or that it cannot name dissolve the portal server. You can start the troubleshooting in the **Network** menu under **Communication** to check that **Gateway** and **DNS** is correctly set.

- **Update denied by server**: the function is activated and the unit had a connection to the server, but the server denied the unit from updating its address. This will happen when you use a service duty where you have to manually add units, for example Comprobo.
- **Internal error**: the function is activated but something unspecified has prevented the device from updating its address in the portal server.

Remember to check that the unit is accessible in both ends. This usually means that you have to open ports in your router or firewall.

15 FAQ's

Q: What can I do if the graphical programming does not start?

A: If the graphical programming doesn't start when you enter the menu, try following steps:

1. Clear the browser's cache. Go to the browser's settings. For firefox you can find this setting under the menu Advanced.

2. Delete the certificate you added as an exception. The procedure differ depending on which browser you are using.

3. Update or reinstall Java. Go to java.com.

16 Technical Specifications UltraBase30

16.1 Universal Inputs

There are 16 universal inputs. These can be used for:

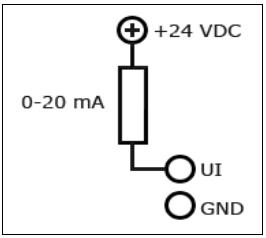
- Analog voltage input: 0-10 V
- Analog current input: 0-24 mA
- Resistance input: 80-249 kΩ (for temperature measurement)
- Digital status: for potential-free sensors

The resistance input has four different areas that can either be selected automatically or selected manually. The areas are 80-249 Ω, 200-2.49 kΩ, 2 kΩ-24,9 kΩ and 20 kΩ-249 kΩ.

The inputs are designed for up to 10 V.

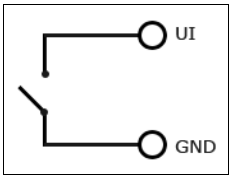
Current measurement:

- Measuring range: 0-24 mA
- Resolution: better than 2 uA
- Accuracy: ± 0,25 %, of full scale
- Input impedance: 235 Ω for current
- Max input signal cont: 30 mA for current



Digital in:

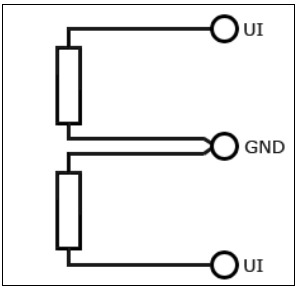
- For short-circuit sensors, on at 150 Ω, off at 200 Ω
- Applied voltage 2 V, short circuit current 1 mA
- Not frequency input
- Not pulse counter
- Min pulse width: 1 sec
- An open input (not connected) is actively low



Voltage measurement:

- Input impedance: 210 kohm
- Accuracy: 100 mV
- Measuring range: 0-10 V

- Resolution: Better than 0.4 mV



Resistance measurement:

- Measuring range: 80 ohm-200 kOhm

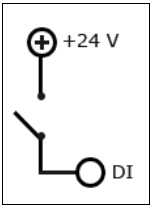
16.1.1 Accuracy

Accuracy at ambient temperature			
Resistance	**-20 to +50 °C**	**+10 to +30 °C**	**Comment**
Measuring range 80 Ω – 240 Ω			
80 Ω	< ±90 mΩ	< ±70 mΩ	
100 Ω	< ±125 mΩ	< ±100 mΩ	~0.25 °C for Pt100
240 Ω	< ±530 mΩ	< ±370 mΩ	
Measuring range 200 Ω – 2.4 kΩ			
200 Ω	< ±0.6 Ω	< ±0.6 Ω	
1000 Ω	< ±1 Ω	< ±0.76 Ω	~0.2 °C for Pt1000
2400 Ω	< ±3 Ω	< ±1.5 Ω	
Measuring range 2 kΩ – 24 kΩ			
2 kΩ	< ±5 Ω	< ±5 Ω	
10 kΩ	< ±12 Ω	< ±10 Ω	
24 kΩ	< ±47 Ω	< ±35 Ω	
Measuring range 20 kΩ – 200 kΩ			
20 kΩ	< ±65 Ω	< ±64 Ω	
100 kΩ	< ±530 Ω	< ±520 Ω	
198 kΩ	< ±2000 Ω	< ±1970 Ω	

16.2 Digital Inputs

There are 4 digital inputs.

All digital inputs have the following specifications:



- Input resistance: 10 kΩ
- Digital input: Switch level from below 3 V to over 4 V
- Pulse counter: Max 200 Hz, at least 2.5 ms pulse width
- Frequence input: 0-200 Hz, 2.5 ms pulse width
- An open input (not connected) is actively low

16.3 Analog outputs

There are 6 analog outputs. Their specifications are:

- Output signal: 0-10 V
- Resolution: 10 bits
- Max output current: 2 mA
- Accuracy: -+100 mV

16.4 Digital outputs

There are 4 digital outputs. Their specifications are:

- Output: Open drain
- Max current: 0,4 A
- Max voltage: 36 V

17 Term Definitions

Acknowledgment: a confirmation that you have received and noticed an alarm.

Alarm: an alarm is always created based on a channel and monitors the channel's value according to set conditions and limits. **A-alarm** has a red indicator and **B-alarm** has a yellow indicator. In addition to these, you can choose to give an alarm a priority from **C-Z** and they have a blue indication.

Alarm groups: manages alarm sendouts and makes it possible to create features for multiple alarms at the same time.

Application: builds the functions in the Ultra. An application contains graphical programming or script and resources such as channels, parameters, curves, all of which are visible in the graphical programming. Overviews and summaries are part of an application. An application can contain other applications, which are **subapplications**. In the graphical programming, an application is represented as a box with inputs and outputs. You can create an application template, which can be useful if you want to create similar applications.

Application backup: a backup with application templates.

Channels: handles all the variable data in the unit. Channels are viewed in overviews and summaries and are logged and monitored by alarms.

Coils: this is a type of a Modbus register. One coil is a 1 bit word.

Components: primitive function blocks in the graphical programming. Connect these with other resources in order to create programs. They exist only in the graphical programming and cannot be reached from other parts of the system.

Curves: a table that is presented as a configurable curve. Curves are used by controllers to do things like converting an outdoor temperature into a flow temperature.

Events/Errors: a type of alarm that is created by the system. It notifies you when something is wrong in the system that may prevent the unit from functioning.

Expansion modules: used to add more inputs and outputs with different features and communicates automatically with the Ultra. You can connect an expansion module and the system will recognize it and its features.

External units: different types of units that an Ultra can exchange information with, as a master. They can be used for more inputs/outputs, meters and more. For example, Modbus and M-bus units can be connected as external units.

File manager: tool that uploads files to the Ultra. There are predefined folders including sensor definitions, communication definitions and backups.

Full backup: a backup containing all of the system's settings and databases, including network settings and users.

Graphical programming: a tool that creates graphical programs. It enables ypu tp build up an entire system with all of its functions based on the inputs and outputs of the unit.

Holiday catalogue: a function usable when working with time schedules. This takes into account weekdays that count as Saturdays and Sundays due to holidays.

I/O-channels: software representations of real inputs and outputs, and allows you to set scale factors, offset and conversion functions for sensors and actuators.

Logs: store values from selected channels with set intervals. Logged data can be displayed as a plot or exported as a table.

Messages: created by script and is stored in the alarm history. An example of a message is that the system has changed into summer operation.

Modbus: a protocol that is used to communicate between different units. Values are ordered in numbered registers, which can be read and written by a master.

Node tree: contains all the elements in the system arranged in a hierarchical order. If you open an application, you can view what's in it, and further, if you fold out a subapplication, you can view what's in the subapplication.

Overview: provides a quick overview of the plant. You can add present

values, alarms and more to this picture. It also allows easy resource editing.

Parameters: store values that are named and set by the user.

Periodic log: this is a log that measures with certain intervals (1 second, 2 seconds, 15 seconds, 1 minute, 5 minutes, 15 minutes, 1 hour or 12 hours). These can measure momentary values, mean value, max value, min value or a sum of a certain period.

Register: a type of Modbus register; one register is a 16 bits word.

Sensors and actuators: physical units that are connected to inputs and outputs.

Smart log: a log that compresses the data that comes in. Second values are converted to minute values and so on. This allows you to store data for a longer period.

Spider view: an overview where you can edit all of the unit's inputs and outputs.

Summary: a view with selected channels, parameters, alarms, alarm groups, curves, time schedules and databases. Here you can see their values and change their settings.

Time schedules: used to set when something should be active or inactive. You can add rules that can be repeated weekly, monthly, early or with a free adjustable interval. You can also select not to repeat the rule at all or only set for a specific date.

Universal input: inputs that measure different kind of quantities (resistance, current and voltage).

Virtual ultra: a simulated unit. Can be used if you want to preconfigure a system.